



Konzeption und Implementierung einer Visualisierung von Simulationsdaten

BACHELORARBEIT

für die Prüfung zum
Bachelor of Engineering

des Studienganges Informationstechnik
an der dualen Hochschule Baden-Württemberg Mannheim

von

Sophia Veronika Schlegel

17. September 2012

Bearbeitungszeitraum

Kurs

Ausbildungsfirma

Betreuer der Ausbildungsfirma

Gutachter der Dualen Hochschule

12 Wochen

415628, TIT09ANS

Deutsches Zentrum für Luft- und Raum-
fahrt e.V., Köln

Dipl.-Math. Sascha Zur

Prof. Dr. Rainer Colgen

Erklärung

Hiermit versichere ich an Eides statt, dass die vorliegende Arbeit von mir persönlich ohne Hilfe Dritter verfasst ist und dass ich keine weiteren als die angegebenen Hilfsmittel und Quellen benutzt habe. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen gedruckter oder elektronischer Form sind entsprechend gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Dasselbe gilt für graphische Darstellungen und Bilder, sowie alle Internetquellen.

Diese Arbeit ist weder ganz noch in Teilen einer anderen Prüfungsbehörde als Leistungsnachweis vorgelegt worden.

Köln, den 17. September 2012

Kurzfassung

RCE (Remote Component Environment) ist ein Simulationsframework zur verteilten Durchführung von Simulationen. RCE wird im Deutschen Zentrum für Luft- und Raumfahrt e.V. speziell für dessen Wissenschaftler entwickelt.

Die mit RCE durchgeführten Simulationen liefern Daten, die von den Wissenschaftlern ausgewertet werden. Eine Auswertung dieser Daten ist jedoch ohne visuelle Aufbereitung und Präsentation nur schwer möglich. Deshalb wird im Zuge dieser Bachelorarbeit ein einheitliches Visualisierungskonzept entwickelt und Möglichkeiten der Umsetzung diskutiert.

Mit der Verwendung aktueller Methoden und Vorgehensweisen sind Grundlagen und verwandte Themen beleuchtet und Anforderungen von Wissenschaftlern und Entwicklern zusammengetragen und analysiert worden. Anhand dieser Grundlagen ist ein umfassendes Konzept zur Visualisierung der Simulationsdaten erstellt und die Art und Weise der Umsetzung umrissen worden.

Die Ergebnisse dieser Abschlussarbeit bilden den Grundstein der RCE-Visualisierung und leiten ein neues Kapitel der RCE-Software ein.

Abstract

RCE (Remote Component Environment) is a framework for distributed simulations in a scientific context. RCE is developed by software engineers of the German Aerospace Center and is especially designed to meet its scientists requirements.

Simulations in RCE produce data, which scientists need to evaluate. This evaluation is hard to accomplish without visual preprocessing. Therefore, this bachelor thesis aims for the improvement of RCE by developing a consistent visualization concept and analyzing possible implementation aspects.

By using state of the art methods und proceedings, basics and related work is defined and requirements of both - developers and scientists - are aggregated. With this knowledge a profound concept for visualizing simulation data has been constructed and the implementation has been outlined.

The results of this thesis is the foundation of visualizing data in RCE and introduces a new chapter of the RCE development.

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
1.1 Aufgabenumfeld	1
1.2 Motivation für eine Verbesserung der Visualisierung in RCE	2
1.3 Aufgabenstellung	3
1.4 Herangehensweise, Entwicklungsumgebung und Gliederung	3
2 Grundlagen	5
2.1 Grundsätze der Daten-Visualisierung	5
2.2 Übersicht über die RCE-Software	6
2.2.1 Die Eclipse Rich Client Platform	6
2.2.2 Aussehen, Architektur und Arbeitsweise der RCE-Software	9
2.2.3 Das Graphical Editing Framework GEF	10
2.3 Aktuelle Visualisierung in der RCE-Software	11
2.4 Visualisierung von wissenschaftlichen Daten in PHX ModelCenter	13
2.4.1 Abgrenzung von RCE-Software und PHX ModelCenter	13
2.4.2 Integration der Visualisierung in Model Center: Der Data Explorer . .	15
2.4.3 Visualisierungsformen in ModelCenter	18
2.5 Visualisierung von wissenschaftlichen Daten in anderen Software-Produkten .	22
3 Anforderungsanalyse	24
3.1 Vorgehensweise	24
3.2 Einsatzszenarios und angestrebte Verbesserungen	25
3.3 Funktionale Anforderungen	28
3.4 Nichtfunktionale Anforderungen	29
3.5 Rahmenbedingungen	30

4 Entwurf eines Visualisierungskonzeptes	32
4.1 Abläufe und Aufmachung der Visualisierung	32
4.1.1 Ablauf der Visualisierung von Daten in der Oberfläche	32
4.1.2 Layout und Struktur der Visualisierung	34
4.2 Funktion und Inhalt der einzelnen Elemente der Visualisierungsoberfläche . .	37
4.2.1 Inhalt der Werkzeug- und Menüleiste	37
4.2.2 Funktionsbereich und Funktionalitäten	39
4.2.3 Graphen, Kurven, Tabellen, Diagramme	40
4.3 Schnittstelle zwischen Workflow und Visualisierung	40
4.4 Separates Visualisierungsfenster vs. Visualisierung mit gegebenen Eclipse-RCP-Mechanismen	41
5 Prototypische Implementierung der Visualisierungsoberfläche zu Vorführzwecken	42
5.1 RCE-Visualisierung mit ausgegliederter Oberfläche	43
5.2 RCE-Visualisierung mit Eclipse-RCP integrierter Oberfläche	44
5.3 Gegenüberstellung von ausgegliederter und integrierter RCE-Visualisierung .	48
6 Fazit	52
Literatur	54
Anhang	56
A. Fragebogen zur Anforderungsanalyse	56
B. Zusammenfassung über den Input der RCE-Benutzer zur Visualisierung	57
C. Übersicht über elektronische Anhänge	58

Abkürzungsverzeichnis

BRD	Bundesrepublik Deutschland
CPACS	Common Parametric Aircraft Configuration Schema
CSV	Comma Separated Values
DLR	Deutsches Zentrum für Luft- und Raumfahrt e. V.
EPL	Eclipse Public License
ESA	European Space Agency
GEF	Graphical Editing Framework
IDE	Integrated Development Environment
JDT	Java Development Tooling
MVC	Model View Controller
OSGi	Open Services Gateway initiative
PDE	Plug-In Developer Environment
RCE	Remote Component Environment
RCP	Rich Client Platform
SC	Einrichtung für Simulations- und Softwaretechnik
SC-VK	Einrichtung für Simulations- und Softwaretechnik, Abteilung für Verteilte Systeme und Komponentensoftware
SCAI	Fraunhofer Institut für Algorithmen und Wissenschaftliches Rechnen
SWT	Standard Widget Toolkit

Abbildungsverzeichnis

1	Architektur der Eclipse-Plattform mit Plugins	8
2	Bestehende Visualisierungsmöglichkeiten in der RCE-Software	12
3	Die Darstellung des Model in der ModelCenter-Software	14
4	Vergrößerung der Werkzeugleiste des Visualisierungstools	16
5	Der Data Explorer - Visualisierungswerkzeug von PHX Model Center	17
6	Contour Plot Ansicht in PHX ModelCenter	21
7	Standardablauf einer Workflow-Visualisierung	33
8	Konzept der RCE-Visualisierung mit einzelnen Elementen	36
9	Screenshot der ausgegliederten Oberfläche	45
10	Screenshot der Umsetzung der RCE-Visualisierung mit integrierter Oberfläche	47

Tabellenverzeichnis

1	User Stories für die Darstellung eines Workflows	26
2	User Stories für die Visualisierung von komponenten- und workflowspezifische Daten	27
3	User Stories für die Darstellung der Daten zur Laufzeit, Nachbearbeitung und Auswertung	28
4	User Stories für die Nachbearbeitung und Auswertung der Simulation und der Daten	29
5	Übersicht über Vor- und Nachteile der vorgestellten Prototypen	50

1 Einleitung

Dieses erste Kapitel dient zur Einführung in die Thematik der Bachelorarbeit und zur Vorstellung der Aufgabenstellung. Zusätzlich werden die Beweggründe für eine Verbesserung der Visualisierungsmöglichkeiten für die Software RCE dargestellt und abschließend eine kurze Erläuterung der Herangehensweise und der Gliederung dieser Bachelorarbeit gegeben.

1.1 Aufgabenumfeld

Die vorliegende Bachelorarbeit ist im DLR in der Einrichtung für Simulations- und Softwaretechnik, Abteilung für Verteilte Systeme und Komponentensoftware (SC-VK) entstanden und bezieht sich inhaltlich auf die RCE-Software (Remote Component Environment).

Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) Das DLR ist die von der Bundesrepublik Deutschland (BRD) beauftragte nationale Raumfahrtagentur und Mitglied der European Space Agency (ESA), dem Verbund der nationalen Raumfahrtagenturen Europas. Im Auftrag der BRD ist es für die Planung und Umsetzung der deutschen Raumfahrtaktivitäten verantwortlich. Zudem ist es eine renommierte deutsche Forschungseinrichtung im Bereich Luft- und Raumfahrt, Energie, Verkehr und Sicherheit.

Das DLR beschäftigt aktuell ca. 7000 Mitarbeiterinnen und Mitarbeiter in 32 Instituten und Einrichtungen an 16 nationalen Standorten und drei internationalen Büros. ¹

Einrichtung für Simulations- und Softwaretechnik (SC) Diese Bachelorarbeit ist organisatorisch in die SC-VK einzuordnen. Der Aufgabenschwerpunkt dieser Einrichtung und insbesondere der Abteilung Verteilte Systeme und Komponentensoftware liegt in der internationalen Softwareforschung und -entwicklung. Hierbei werden Softwareprojekte aus zahlreichen Bereichen der Wissenschaft bearbeitet. Beispiele hierfür sind das Management wissenschaftlicher Daten, Software in verteilten Systemen und Software für das Simulations- und Workflow-Management. Zudem unterstützt diese Einrichtung die Softwareentwickler des DLR bei der Einführung moderner Software-Engineering-Techniken und -Methoden und hat so eine Vorbild- und Vorreiterrolle innerhalb des DLR. ²

¹vgl. DLR (2011): Das DLR im Überblick. Abs. 1, 5

²vgl. DLR (2011): Simulations- und Softwaretechnik. Verteilte Systeme und Komponentensoftware. Abs. 1

Remote Component Environment (RCE) Die RCE-Software ist ein Produkt der Einrichtung für Simulations- und Softwaretechnik, Abteilung für Verteilte Systeme und Komponentensoftware. Innerhalb dieser Umgebung arbeiten mehrere Entwickler an unterschiedlichen Bereichen von RCE. Die RCE-Software wird in erster Linie für Wissenschaftler aus den Bereichen Luftfahrt und Verkehrswesen innerhalb des DLR entwickelt.

1.2 Motivation für eine Verbesserung der Visualisierung in RCE

Bei RCE handelt es sich um ein Software-Produkt, dass speziell für Wissenschaftler innerhalb des DLR entwickelt wird. Aus diesem Grund arbeiten die RCE-Entwickler eng mit den Benutzern zusammen um höchst effizient und zielgerichtet die RCE-Entwicklung voran zu treiben.

Vor der Einführung der RCE-Software waren die Wissenschaftler des DLR auf ModelCenter PHX angewiesen gewesen. Dabei handelt es sich um ein kommerzielles Softwareprodukt aus den USA, das unterschiedlichste Anwendungsbereiche integriert und nur bedingt Anwendungsfälle innerhalb des DLR abdeckt.

RCE ist ein Framework, das zur Konzeption und Durchführung von Simulationen aus unterschiedlichen Bereichen der Luftfahrt und dem Verkehrswesen entwickelt wird. Hierbei fallen Simulationsdaten an, die von den jeweiligen Wissenschaftlern ausgewertet werden müssen. Bisher übernimmt die RCE-Software jedoch nur die Konzeption und Durchführung der Simulationen, die Auswertung der Simulationsdaten ist nicht in der RCE-Software möglich und wird auf unterschiedlichste Weise durchgeführt. Hierzu werden die in RCE entstehenden Daten mithilfe anderer Softwareprodukten visualisiert und, bearbeitet und ausgewertet.

Für die RCE-Software und deren Nutzer ist es wünschenswert, den Schritt der Datenauswertung in den Arbeitsablauf von RCE zu integrieren und somit die Benutzung der RCE-Software zu verbessern. Mit der Ausführung der Simulation soll dann innerhalb von RCE die Überwachung Auswertung der Simulation möglich sein, ohne die Daten außerhalb RCE anderweitig zu verarbeiten.

Ziel dieser Bachelorarbeit ist es, die Visualisierung innerhalb der RCE-Software zu realisieren und Möglichkeiten für eine Umsetzung zu finden. Die RCE-Software soll durch die Erweiterung um die Datenvisualisierung attraktiver gestaltet und die Softwarequalität dadurch verbessert werden.

1.3 Aufgabenstellung

Im Rahmen dieser Bachelorarbeit soll für die Integrationsumgebung RCE eine Visualisierungskomponente für die Daten von berechneten Simulations-, Optimierungs- und Studienergebnissen entwickelt werden. Hierbei sind zu Beginn die bestehenden Möglichkeiten zu erarbeiten und aufbauend auf diesen Ergebnissen ein Visualisierungskonzept für die RCE-Software zu entwickeln. Abgeschlossen wird die Aufgabe mit einer prototypischen Implementierung einer RCE-Visualisierungskomponente.

Das Ziel dieser Arbeit ist es, nach den gängigen Software-Engineering-Prozessen ein Konzept zur Verbesserung der Visualisierung von Daten im RCE-Kontext zu erstellen. Schwerpunkt liegt vor allem auf einer umfangreichen und weitreichenden Anforderungsanalyse und Konzeptentwicklung der RCE-Visualisierung.

1.4 Herangehensweise, Entwicklungsumgebung und Gliederung

Methodisch ist die Arbeit in sechs Schritte unterteilt:

- Literaturrecherche und Aneignen von Grundlagen-Wissen
- Anforderungsanalyse
- Entwicklung eines Visualisierungskonzeptes
- Evaluierung von Umsetzungsmöglichkeiten im RCE-Kontext
- Prototypische Implementierung
- Auswertung der gewonnenen Ergebnisse

Die RCE-Software basiert auf dem Eclipse-Framework. Aus diesem Grund wird die Entwicklungsumgebung Eclipse benutzt und in der Programmiersprache Java entwickelt.

Das sich an dieses Kapitel anschließende Kapitel 'Hintergrund' stellt kurz die theoretischen Grundlagen dar. Hierbei wird vor allem die Arbeitsweise der RCE-Software, die verwandete ModelCenter PHX Software und allgemeine Visualisierungsmöglichkeiten vorgestellt um so dem Leser ein fundiertes Basiswissen zu vermitteln.

Dieses Wissen dient vor allem als Grundlage für die im folgenden Anforderungskapitel durchgeführte Anforderungsanalyse. Anhand der Anforderungen wird ein Visualisierungskonzept

entwickelt und Möglichkeiten der Umsetzung diskutiert. Darauf folgt eine prototypische Implementierung zweier unterschiedlich realisierter Visualisierungsoberflächen. Abschließend erfolgt eine ausführliche Auswertung der Ergebnisse.

2 Grundlagen

In diesem Kapitel wird zunächst eine Herangehensweise und ein Bewertungsschema für die Visualisierung von Daten vorgestellt. Danach werden zunächst kurz die RCE-Software und die für das Verständnis wichtigen Grundlagen erläutert. In diesem Zusammenhang ist kurz die RCE-Basis, die Rich Client Platform Eclipse, umrissen. ModelCenter PHX wird als Konkurrenzprodukt der RCE-Software in die Betrachtungen miteinbezogen. Im Anschluss daran werden interessante Aspekte von anderen Software-Produkten, die nicht im wissenschaftlichen oder technischen Bereich liegen, betrachtet.

2.1 Grundsätze der Daten-Visualisierung

Ein interessanter Ansatz für die grundlegende Herangehensweise von Visualisierungsproblematiken bietet ein Zitat von Ben Shneiderman aus einem Konferenzbeitrag aus dem Jahr 1996:

”A picture is often cited to be worth a thousand of words and, for some (but not all) tasks, it is clear that a visual presentation—such as a map or photograph—is dramatically easier to use than is a textual description or a spoken report.”³

Bereits Shneiderman hat in seinem Konferenzvortrag die Wichtigkeit der Datenvisualisierung thematisiert. Das seiner Arbeit zugrunde liegende Basisprinzip reduziert er auf drei wesentliche Schritte: ”Overview first, zoom and filter, then details-on-demand”.⁴ Dieses erste Basisprinzip wird *Visual Information Seeking Mantra* genannt und im selben Artikel von Shneiderman auf sieben Schritte expandiert:⁵

1. Overview: Überblick geben
2. Zoom: Interessante Punkte detaillieren
3. Filter: Uninteressantes ausfiltern
4. Details-on-demand: Einzelne Punkte auswählen, Details anzeigen
5. Relate: Verbindungen zwischen Punkten erkennen

³vgl. Shneiderman (1996): The Eye Have It: A Task by Data Tyoe Taxonomy for Information Visualization. Absatz 2

⁴vgl. Shneiderman(1996): Absatz 2

⁵vgl. Shneiderman(1996): Absatz 3

6. History: Überblick über gesichtete Informationen Speichern

7. Extract: Exportieren und extrahieren von Informationen

Dieses Prinzip lässt sich auch heute noch einfach und zuverlässig auf die meisten wissenschaftlichen Forschungsbereiche, in denen Datenmengen visualisiert werden sollen, anwenden. In seiner Arbeit nimmt Shneiderman im weiteren Verlauf eine Klassifizierung von Datentypen vor. Diese werden in ein-, zwei-, drei- und mehrdimensionale, zeitliche, baumartige und zusammenhängende Datentypen eingeteilt. Das Grundprinzip der sieben Schritte ist für eine Konzipierung einer Visualisierungskomponente im RCE-Bereich ein einfaches, leicht anzuwendendes Prinzip. Im RCE-Kontext ist die passgenaue Konzipierung für verschiedene Anwendungsfälle jedoch im Vordergrund und eine generelle Einteilung der Datentypen wie in Shneidermans Paper nicht nötig.

2.2 Übersicht über die RCE-Software

Angestoßen durch das Projekt SESIS (Schiffsentwurfs- und Simulationssystem), wird die RCE-Software (Remote Component Environment) im DLR in der Einrichtung SC-VK in Kollaboration mit dem Fraunhofer Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI) entwickelt.⁶ Es handelt sich dabei um ein Simulationsframework auf der Basis der Eclipse Rich Client Platform (RCP) (siehe Kapitel 'Die Eclipse RCP'). Durch dieses Framework können allgemeine und häufig verwendete Softwarekomponenten für die in RCE integrierten Applikationen bereit gestellt werden.

RCE ist plattformunabhängig und dient zur Ausführung von Simulationen in verteilten Systemen. RCE ist unter der Eclipse Public License (EPL) verfügbar und aktuell in der Version 2.3.5 vorhanden.⁷

2.2.1 Die Eclipse Rich Client Platform

Die Rich Client Platform RCP des Eclipse-Projektes ist ein Softwareprojekt unter der Leitung der Eclipse-Foundation. Alle Projekte der Eclipse-Foundation haben das gemeinsame Ziel, eine unabhängige Plattform für unterschiedliche Bereiche der Softwareentwicklung bereitzustellen.

⁶vgl. Fraunhofer SCAI (2012): Produkte. RCE.

⁷vgl. DLR (2011): Simulations- und Softwaretechnik. Software. RCE.

Unter einem Rich Client versteht man eine Software, die sowohl Präsentationsfunktionalitäten, als auch die Anwendungslogik bereitstellt und über Schnittstellen individuell erweitert werden kann. Bei der Eclipse RCP (Rich Client Platform) handelt es sich um ein solches Framework. Hierbei wird es ermöglicht, eine Applikation basierend auf Plugins zu entwickeln. Plugins sind Softwarekomponenten, die beliebig mit anderen Komponenten kombinierbar sind. Somit können auf einer gemeinsamen Basis viele unterschiedliche Anwendungen für die unterschiedlichsten Bereiche entwickelt werden. Die Integration der Plugins geschieht zentral auf der Basis des Open Services Gateway initiative (OSGi)-Frameworks. Die Eclipse-IDE (Integrated Development Environment) ist ein Beispiel für eine Applikation, die mittels Plugins die Eclipse-RCP zu einer Integrated Development Environment erweitert.

Die Eclipse-RCP ist plattformunabhängig und in der Programmiersprache Java als Laufzeitsystem entwickelt. Aus diesem Grund wird sowohl innerhalb von RCE, als auch im Rahmen dieser Bachelorarbeit auf Java-Grundlage entwickelt.⁸

Architektur und Plugin-Konzept

Die Eclipse Plattform besteht zunächst einzig aus dem Laufzeitsystem (Abbildung 1 unten, *Platform Runtime*). Alle weiteren Funktionalitäten werden als Plugin oder als Satz von Plugins zum Kern hinzugefügt und von diesem dementsprechend verarbeitet. Dieses Laufzeitsystem bildet als OSGi-Framework das Grundgerüst der Eclipse Plattform.

Plugins fügen Funktionalitäten unterschiedlichster Art zum Kern hinzu und reichen von der Einbindung anderer Programmiersprachen und Bibliotheken bis hin zu Versionskontrolle und Datenbank-Management-Systemen.

Die Eclipse Plattform wird standardmäßig mit einem vorgefertigten Satz an Plugins ausgeliefert. Hierzu zählen neben der Oberfläche und dem Hilfesystem auch Versionskontrollfunktionen, das Java Development Tooling (JDT) und das Plug-In Developer Environment (PDE). In Abbildung 1 sind dies die Funktionen links und in der Mitte.¹⁰

Die Java-Frameworks Standard Widget Toolkit (SWT) und JFace sind für die Oberfläche der Plattform zuständig. Hierbei werden diese sowohl als einheitliche Schnittstelle für Zusätze zur Benutzeroberfläche als auch für die Oberfläche selbst genutzt.

⁸vgl. Eclipse Foundation (EF) (2012): Eclipsepedia. Main Page/Rich Client Platform.

⁹vgl. EF (2011): Eclipse Documentation. Platform Plug-In Developer Guide. Stand: 26. Juni 2012

¹⁰vgl. EF (2012): Eclipsepedia. Plugin-Development Environment (PDE). Stand: 25. Juni 2012

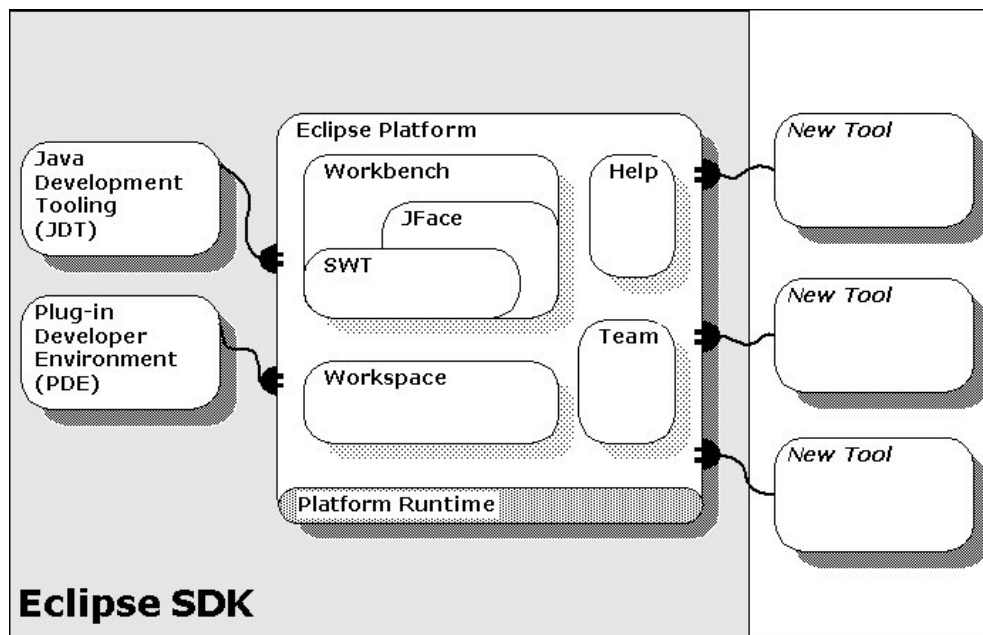


Abbildung 1: Architektur der Eclipse-Plattform mit Plugins ⁹

Graphical User Interface: Workbench, Perspektiven, Views, Editoren

Die graphische Benutzeroberfläche der Eclipse Plattform ist in vier Elemente unterteilt: Workbench, Perspektiven, Editoren, Views. Diese unterliegen einer hierarchischen Gliederung.

Workbench Die Workbench ist das Fenster der eigentlichen Benutzeroberfläche. Wird die Plattform gestartet, öffnet sich zunächst ein Workbench-Fenster. Innerhalb der Workbench befinden sich dann unterschiedliche Perspektiven.

Perspektive Eine Perspektive ist eine Gruppierung von Views und Editoren innerhalb einer Workbench und enthält unterschiedliche Funktionalitäten. Die Perspektive ist auf einen Anwendungsfall zugeschnitten oder arbeitet bezüglich unterschiedlicher Ressourcen, enthält also alle Komponenten, die für die Bearbeitung einer Aufgabe wichtig sind. Perspektiven sind für das Erscheinungsbild der Menus und Toolbars verantwortlich. Innerhalb einer Workbench gibt es eine Reihe von Editoren, auf die die unterschiedlichen Perspektiven zugreifen. Die meisten Perspektiven beinhalten einen einzigen zentralen Editor mit Views zur näheren Beschreibung.

Editoren Unterschiedliche Dateitypen können mit verschiedenen Editoren assoziiert werden, wie zum Beispiel der standardmäßig integrierte Java-Editor in der Workbench. Gibt es keinen passenden Editor zu einer Ressource, wird ein externer Editor geöffnet (zum

Beispiel Microsoft Word für *.doc-files). Unter Windows sind manche externen Editoren in der Workbench integriert (Embedded Editors). Mehrere Editoren können gleichzeitig in einer Workbench geöffnet werden, aber nur einer dieser kann tatsächlich aktiv sein. Je nach Editor enthalten die Menuleisten und Toolbars unterschiedliche Buttons und Kommandos.

Views Zur näheren Beschreibung von Workbench-Inhalten und der in einem Editor geöffneten Ressource gibt es innerhalb einer Perspektive mehrere Views. Diese können eigene Menuleisten und Toolbars beinhalten, die Aktionen die diese ausführen betreffen jedoch nur den in der View angezeigten Inhalt. Der Project Explorer beispielsweise beinhaltet die Projekte und Dokumente des aktuellen Arbeitsumfeldes, die Help-View zeigt die Hilfe zu unterschiedlichen Themen an. Beide Views stellen Funktionen zum Bearbeiten des jeweiligen Inhaltes bereit, wie zum Beispiel das Erstellen eines neuen Projektes (Project Explorer) oder die Suchfunktion in der Hilfe.

Die Workbench beschreibt also das Fenster, das die gesamte Applikation beinhaltet. Perspektiven gruppieren innerhalb der Workbench einen Satz an Views und Editoren.^{11 12}

2.2.2 Aussehen, Architektur und Arbeitsweise der RCE-Software

RCE ist eine Applikation auf Basis der Eclipse Rich Client Platform und dem OSGi-Framework Equinox. Die RCP dient dabei als graphische Benutzerschnittstelle für ein Simulationsframework. Die Software ist als Paket von verschiedenen Eclipse-Plugins konzipiert. Ein RCE-Plugin ist ein Plugin-Project der Eclipse Foundation. Als solches können diese Plugins durch die Mechanismen der RCP in das Grundgerüst eingefügt werden. Die RCE-Plugins sind in verschiedene Bereiche unterteilt. Die Kern-Plugins betreffen alle Kernfunktionalitäten der RCE-Software, wie zum Beispiel die graphischen Elemente (Views, Workflow-Editor), die Authentifizierung und das Datenmanagement. Grundsätzlich enthält RCE zusätzlich zum Kern dann noch die Plugins für die Komponenten eines Workflows und allgemeine Konfigurationsplugins.

Equinox dient hierbei als Modularisierungsspezifikation und wird zur Trennung der Laufzeitumgebung von graphischer Oberfläche und anderen Komponenten benötigt.

¹¹vgl. Hille (2009): Eclipse Rich Client Platform (RCP). S. 5

¹²vgl. EF (2011): Eclipse Documentation. Platform Plug-in Developer Guide. Stand: 29. Juni 2012

Im Mittelpunkt der RCE-Oberfläche steht der für die Visualisierung von Simulationen entwickelte Workflow-Editor. Mithilfe dieses Editors können die einzelnen Komponenten einer Simulation als graphische Elemente dargestellt und untereinander verbunden werden. Eingangs- und Ausgangsdaten mit zugehörigen Datentypen einer Komponente können dabei ebenfalls spezifiziert werden. Wird ein Workflow simuliert, liefern die Komponenten dann die eigentlichen Ergebnisse zu den spezifizierten Ein- und Ausgangsdaten. Zur Anzeige dieser Daten gibt es dann noch weitere Views, wie der Workflow Data Browser, die Workflow Console oder komponentenspezifische Views wie die Parametric Study View. Diese graphischen Elemente bilden die RCE-Perspektive.

Der Workflow-Editor und die verschiedenen Views bilden die bisherige Grundlage für die Visualisierung der Eingangs- und Ausgangsdaten, der Workflows und deren Beziehungen zueinander.

Für eine mögliche Umsetzung der Visualisierung im RCE-Kontext ergibt sich somit auch die Frage, ob die Visualisierung als eigenständige Komponente, als Zusatz zur bestehenden Oberflächenkomponente oder als externes Werkzeug (als externe Applikation) in die RCE-Software integriert wird.

2.2.3 Das Graphical Editing Framework GEF

Das Graphical Editing Framework GEF wird, wie auch die Rich Client Platform selbst, von der Eclipse Foundation entwickelt. GEF dient dazu, einfach graphische Editoren und Views für die Eclipse RCP herzustellen. GEF beruht auf SWT und besteht aus drei Komponenten:

Draw2d: Draw2d bietet Funktionen zur Anzeige und Integration von grafischen Komponenten auf einem SWT Canvas-Element an. Grafiken (= Figures) werden als simple Java-Objekte betrachtet, gängige Listener für SWT Events sind dabei schon vordefiniert. Draw2d kann sowohl außerhalb von Eclipse, als auch in Kombination mit GEF oder Zest Komponenten genutzt werden.

GEF (MVC): Mit GEF (MVC) wird zu SWT und Draw2d Elementen Funktionalitäten für Bearbeitung und Editierbarkeit hinzugefügt. GEF (MVC) ist ein Model-View-Controller Framework mit dem eigene Editoren für die Eclipse RCP hergestellt werden können.

Zest: Bei Zest handelt es sich um das Eclipse Visualization Toolkit. Zest ist nahtlos inner-

halb Eclipse nutzbar, da es ausschließlich mit SWT und Draw2d entwickelt worden ist. Mit Zest können grafische RCP Views erstellt und in die Workbench integriert werden. Zudem beinhaltet Zest ein Layout-Paket für Graphen.

Bei dem RCE-Workflow-Editor handelt es sich bereits um einen Editor auf der Basis von GEF (MVC) und Draw2d. Dieser ist nahtlos in die Eclipse RCP integriert. Aus diesem Grund bietet sich eine Benutzung von GEF für die RCE-Visualisierung an.

2.3 Aktuelle Visualisierung in der RCE-Software

Die "Gebrauchstauglichkeit" (engl. Usability) eines Softwareproduktes ist ein wesentliches Qualitätsmerkmal eines Systems. Demnach ist es sowohl für den Benutzer, als auch für den Entwickler äußerst wichtig, eine benutzerfreundliche Arbeitsoberfläche in dem Software-Produkt bereitzustellen.

Um große wissenschaftliche Datenmengen und deren Beziehungen zueinander sinnvoll und übersichtlich darzustellen, ist die Visualisierung, insbesondere mittels Graphen, Diagrammen, Tabellen und Modellen unumgänglich. Die RCE-Software ist ein Simulationsframework, das solche wissenschaftlichen Daten produziert.

Innerhalb der RCE-Software gibt es bereits mehrere Visualisierungsansätze und -möglichkeiten, die jedoch sowohl in ihrem Funktionsumfang, als auch in ihrer Benutzbarkeit verbessert werden können.

Für die Zusammenstellung und Ausführung der einzelnen Workflows dient der Workflow-Editor. Dieser ist in Abbildung 2 im oberen Teil zu sehen. Die Simulation durchläuft dabei zunächst die Optimierungskomponente und wird anschließend in der Pythonkomponente modifiziert.

In diesem Workflow-Editor sind die einzelnen Simulationskomponenten als klick- und auswählbare Icons, die mittels Pfeilen verbunden werden können, dargestellt. Die Eingangs- und Ausgangsdaten der Komponenten können in zusätzlichen Workflow-Views editiert und angepasst werden.

Für die Optimiererkomponente existiert in der Optimizer-View eine Darstellungsmöglichkeit der Ein- und Ausgangsdaten in Tabellenform. Zusätzlich können die Daten in einem Koordinatensystem als Kurve graphisch dargestellt werden. Diese Optimierer-View ist im unteren

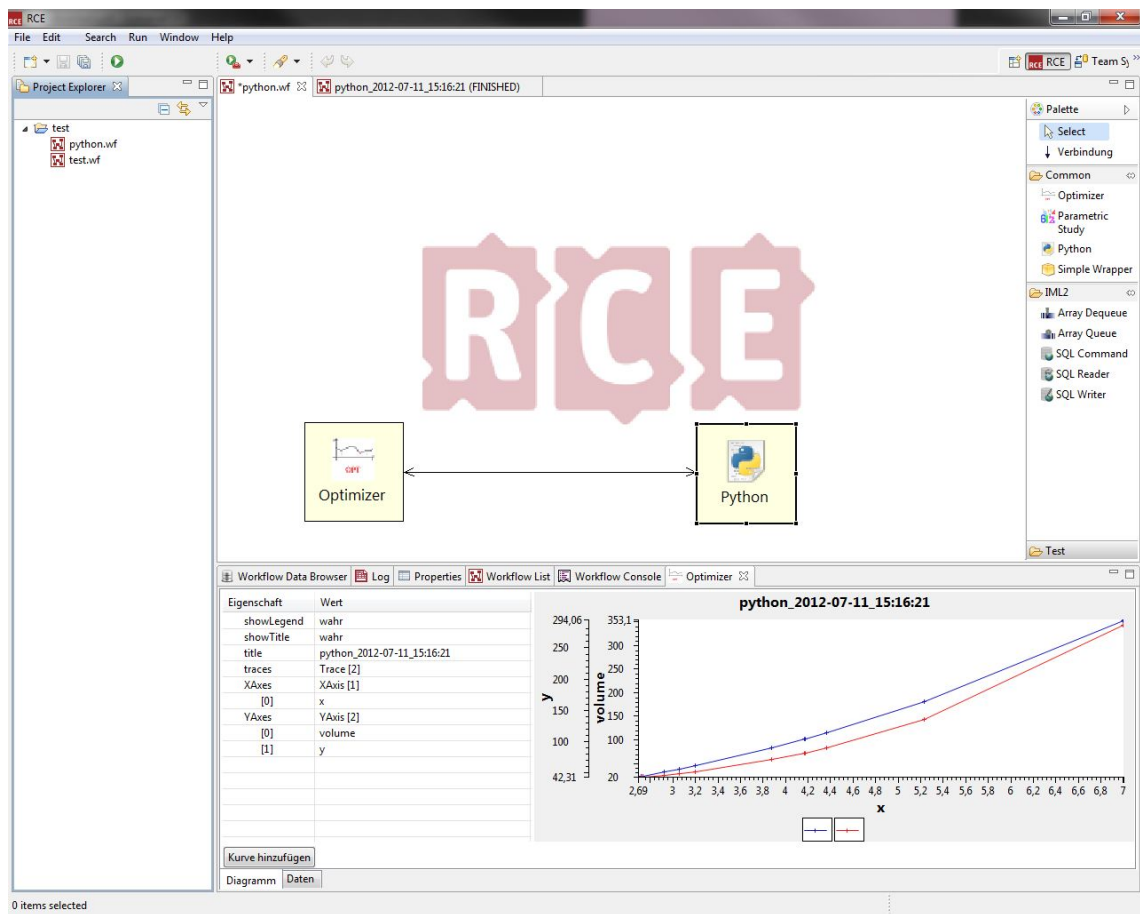


Abbildung 2: Bestehende Visualisierungsmöglichkeiten in der RCE-Software

Teil von Abbildung 2 zu finden.

Zusätzlich gibt es in einer speziellen RCE-Implementierung eine Darstellungsmöglichkeit für Common Parametric Aircraft Configuration Schema (CPACS)-Daten. Diese Visualisierungsmöglichkeit betrifft jedoch zunächst nicht eine allgemeine RCE-Visualisierungskomponente.

Jede Komponente in RCE besitzt Eingangs- und Ausgangsdaten, die graphisch dargestellt werden können und sollten. In der bisherigen RCE-Software werden Daten jedoch bisher nur marginal und in kleinem Umfang umgesetzt. Prinzipiell könnte zwar für jede Komponente eine View mit graphischer Darstellung ähnlich der Optimierer-View implementiert werden. Dies führt jedoch zwingend zu Codedopplungen. Zudem gibt es weitaus mehr Funktionalitäten als nur eine graphische Darstellung, um die Daten darzustellen. Eine allgemeine und stimmige Aufmachung der Visualisierung ist außerdem wünschenswert und würde die Usability der RCE-Software wesentlich verbessern.

PHX ModelCenter ist vor der RCE-Software zur Konzeption und Entwicklung von Simulationen genutzt worden. Aus diesem Grund wird die ModelCenter im nächste Kapitel näher betrachtet.

2.4 Visualisierung von wissenschaftlichen Daten in PHX ModelCenter

PHX ModelCenter ist ein kommerzielles Software-Produkt der Firma Phoenix Integration und dient der Optimierung, Integration und Automatisierung von Engineering-Prozessen und -Systemen. Hauptaufgabe von ModelCenter ist die Kosten- und Zeiteinsparung und die Vereinfachung des Arbeitsprozesses während einer Produktentwicklung in den verschiedensten Ingenieursbereichen.

2.4.1 Abgrenzung von RCE-Software und PHX ModelCenter

Mittels Computersimulationen können virtuelle Prototypen eines Produktes hergestellt werden und somit viele Design- und Architekturfragen geklärt werden. Diesen Prozess unterstützt ModelCenter.

Zur Erweiterung der ModelCenter-Software gibt es Erweiterungspacks, die dann ModelCenter an die verschiedenen Aufgaben im Detail anpasst. Von besonderem Interesse ist hier das VisualizationPack ModelCenter ist für das Betriebssystem Microsoft Windows entwickelt, die aktuelle Version ist 10.0.¹³

ModelCenter ist unter einer proprietären Lizenz erhältlich, was mitunter ein Grund für die Entwicklung der RCE-Software gewesen ist. Zudem ist ModelCenter eine Standardsoftware und nicht auf den Gebrauch im DLR spezifiziert, zum anderen ist eine Interaktion mit den ModelCenter-Entwicklern nicht möglich um etwaige Anforderungen und Vorschläge zu kommunizieren. Aus dieser Situation heraus ist die RCE-Software entstanden, die an den jeweiligen Einsatzbereich angepasst werden kann.

Wie auch in RCE gibt es in ModelCenter eine Ansicht, die die gesamte Simulation darstellt. Das Pendant zum Workflow in der RCE-Software ist das Model in ModelCenter. Der Model-Editor ist in Abbildung 3 dargestellt (blauer Hintergrund). Ein Model besteht aus Komponenten und deren Verbindungen untereinander. Zusätzlich können in das Model Kom-

¹³vgl. Phoenix Integration (2012): PHX ModelCenter - Desktop Trade Studies. Stand 17. Juli 2012

mentare, Bilder, Macros und Datenmonitore eingefügt werden, die die eigentliche Simulation überwachen.

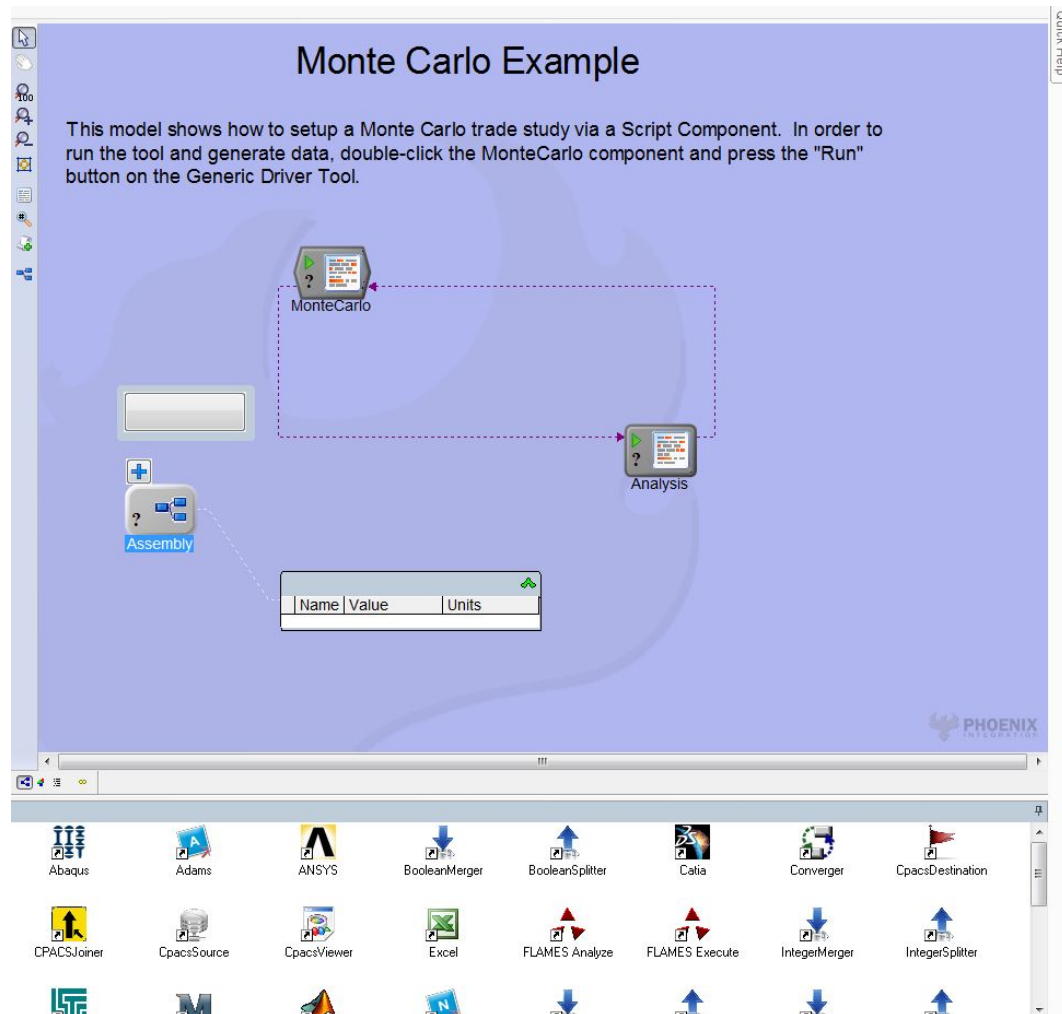


Abbildung 3: Die Darstellung des Model in der ModelCenter-Software

Neben gängigen Komponenten wie der Excel-Komponente und einer Optimiererkomponente, die auch in die RCE-Software integriert sind, bietet ModelCenter viele weitere Komponenten in verschiedenen Bereichen an (s. Abbildung 3, unten). So können mit der Matlab-Komponente eigene Matlab-Dateien in das Model integriert werden oder mit diversen CPACS-Komponenten CPACS-Datensets (CPACS = Common Parametric Aircraft Configuration Schema) verarbeitet werden.

Mit einem Klick auf die jeweilige Komponente wird die Komponentenkonfiguration geöffnet (in separatem Fenster). Wird dann die jeweilige Komponente ausgeführt, öffnet sich automatisch die Visualisierungsansicht, der Data Explorer (s. Kapitel 'Integration der Visualisierung

in Model Center: Der Data Explorer’).

Grundsätzlich können sowohl in der RCE-Software, als auch in ModelCenter, die anfallenden Daten Ein- oder Ausgänge eines gesamten Workflows (bzw. Model in ModelCenter) oder einer einzelnen Komponente aus einem Workflow darstellen. In ModelCenter ist es grundsätzlich nur möglich, einzelne Komponenten und deren Daten zu visualisieren. Um Input und Output eines gesamten Models darzustellen, müssen die Komponenten zusammengeschlossen und nach außen als Komponente gekapselt werden. Diese Kapselung wird in ModelCenter über eine sogenannte *Assembly* realisiert.

In der RCE-Software ist ein solcher Gruppierungsmechanismus für Komponenten bereits angedacht und im Konsens beschlossen. Die vorliegende Arbeit zielt zunächst nur auf Konzepterstellung und prototypische Implementierungen ab. Deswegen kann die Kapselung von Komponenten in einem Komponentencontainer der Visualisierung zugrunde gelegt werden.

2.4.2 Integration der Visualisierung in Model Center: Der Data Explorer

Neben dem Workflow-Editor gibt es in PHX ModelCenter einen umfangreichen Visualisierungsteil für die anfallenden Simulationsdaten. Dieser Visualisierungsteil ist der Data Explorer und ist für die Studien- und Forschungswerkzeuge in ModelCenter entwickelt. Solche Werkzeuge sind unter anderem die Parameterstudie, Design of Experiments oder Optimiererwerkzeuge, die zum Teil auch in RCE zu finden oder in Zukunft angestrebt sind. Über diese Visualisierungskomponente können zur Laufzeit und auch im Anschluss einer Simulation des Models die anfallenden Daten gesammelt und angezeigt werden.

Der Data Explorer von ModelCenter ist in Abbildung 5 dargestellt. Der Data Explorer lässt sich grob in vier Elemente aufteilen: Die Werkzeugleisten und Toolbars (s. Abbildung 4, oben und Abbildung 5, oben), eine Übersicht über geöffnete Visualisierungen und Plots (s. Abbildung 5, linker Rand), die aktive Visualisierungsansicht (s. Abbildung 5, Hauptteil) und die Menüleiste dieser aktiven Visualisierung (s. Abbildung 5, rechter Rand).

Zur Laufzeit gibt es eine Fortschrittsanzeige mit den bereits erhaltenen Daten und eine Möglichkeit zur Unterbrechung und Wiederaufnahme der Simulation (s. Abbildung 4, rechts). Neben diesen Funktionalitäten bietet die ModelCenter-Visualisierung zusätzlich Funktionen zum Drucken, Exportieren und Speichern an (s. Abbildung 4, links).

Neben diesen Standardfunktionalitäten besitzt ModelCenter die Möglichkeit, externe Daten

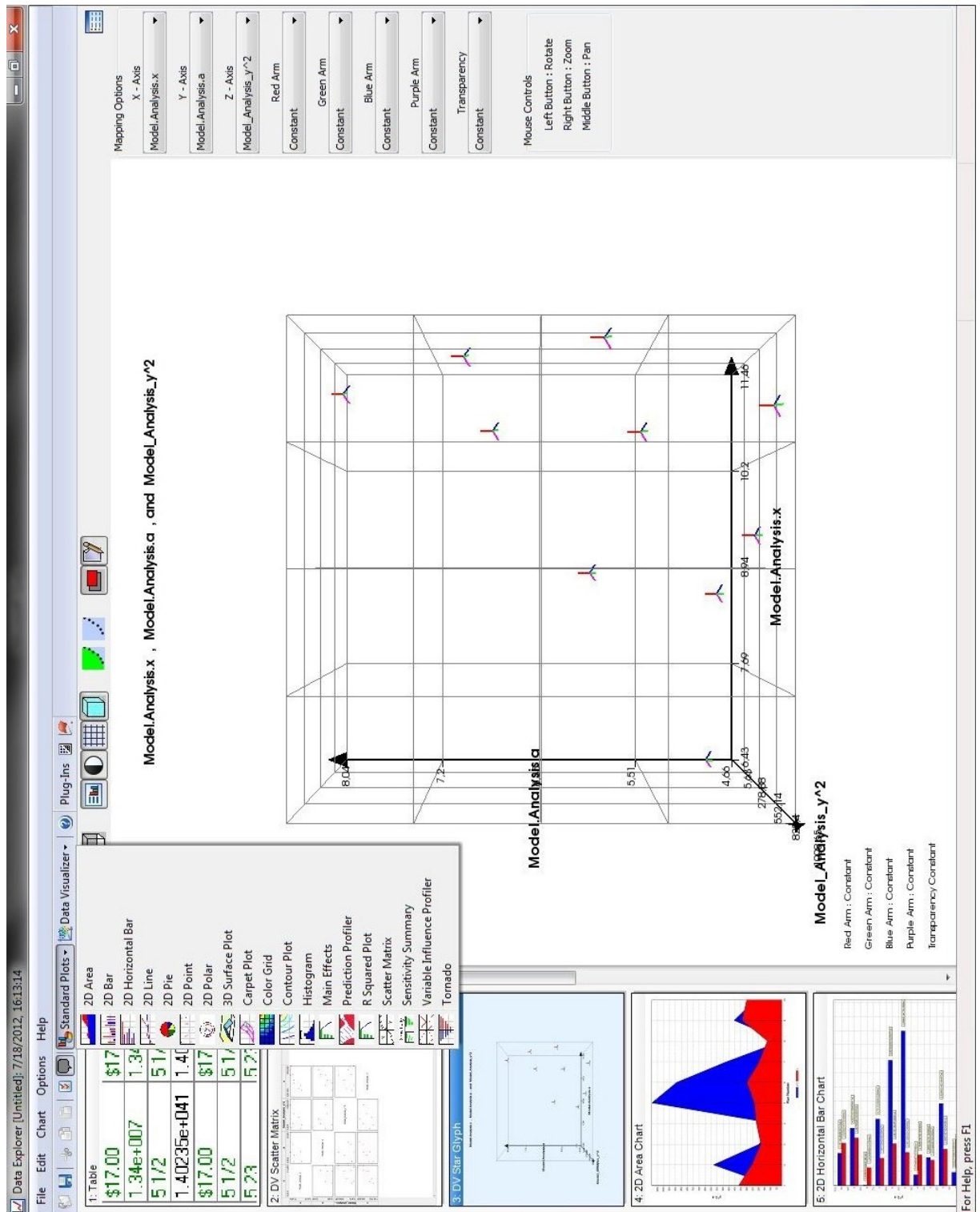


Abbildung 5: Der Data Explorer - Visualisierungswerkzeug von PHX Model Center

2.4.3 Visualisierungsformen in ModelCenter

ModelCenter bietet unterschiedliche Visualisierungsformen und Plots für unterschiedliche Arbeitsbereiche an. Diese sind in Standarddarstellungen und spezielle Datenvisualisierungen unterteilt. Im Folgenden sind die einzelnen Visualisierungsmöglichkeiten dieser beiden Bereiche aufgelistet.

Standardvisualisierungen

Die Standardvisualisierungen beinhalten alle Basisdarstellungsoptionen. Diese sind im Grundpaket von PHX ModelCenter bereits enthalten.

Tabellenansicht Die Tabellenansicht ist die Default-Ansicht für jede Simulation und die Grundlage für jede weitere Visualisierung.

2D Point/Line/Area/Polar Chart Bei diesen Darstellungsformen werden die gemessenen Werte in Relation zum Iterationsschritt in einem Koordinatensystem dargestellt, als Punkte, als Kurve, als Fläche unter der Kurve oder als Linie in einem kreisförmigen Koordinatensystem.

2D Bar Chart, Pie Chart Hierbei handelt es sich um Diagramme zur Darstellung von Verteilungen auftretender Werte. Das Säulendiagramm stellt den Wert je Iteration als Balken dar (vertikal oder horizontal), das klassische Kuchendiagramm zeigt die Häufigkeit eines Wertes anteilig von 100% der Werte.

3D Surface Plot, Carpet/Contour Plot In diesen Visualisierungsformen wird die Abhängigkeit eines Wertes/einer Variablen von zwei weiteren dargestellt. Der Surface Plot ist hierbei das klassische dreidimensionale Koordinatensystem in dem die Werte als Ebene dargestellt werden. Der Carpet Plot stellt diese drei Variable im zweidimensionalen dar um zusätzliche Inter- und Extrapolationen zu vereinfachen. Der Contour Plot stellt die Werte ebenfalls zweidimensional dar, stellt dabei die abhängige Farbe aber unterschiedliche gefärbt dar.

Histogramme Mit der Histogrammansicht können gezielt statistische Auswertungen der Simulationen vorgenommen werden. Neben dem Histogramm gibt es eine Übersicht über statistische Daten, wie der Median, die Standardabweichung und das arithmetische Mittel.

Variable Influence/Prediction Profiler Diese Werkzeuge dienen dazu, die Werte von Ausgangsvariablen abzuschätzen in Abhängigkeit von anderen Variablen. Dargestellt werden diese dann in kartesischen Koordinatensystemen oder Contour Plots. Mit dem Prediction Profiler wird diese Abschätzung getätigt, mit dem Variable Influence Profiler können einzelnen Variablen zusätzlich priorisiert werden und so Probleme zu vereinfachen.

Scatter Matrix Bei einem Scatterplot oder Streudiagramm handelt es sich um eine Darstellung von beobachteten Wertepaaren von Merkmalen, deren Zusammenhang unklar ist und erfasst werden soll. Diese Paare werden als Punkte in einem Koordinatensystem repräsentiert um so einen Zusammenhang, eine Tendenz oder ähnliches festzustellen.

Sensitivity Plots Mit dem Sensitivity Summary kann der Grad der Beeinflussung von Variablen auf den Ausgang einer Simulation festgestellt und statistisch ausgewertet werden. Der Tornado Plot und der R Squared Plot ist die gezielte Auswertung einer einzelnen Variablen auf einen bestimmten Ausgangswert. Der Main Effects Plot dient zur Darstellung aller Auswirkungen auf die Ausgangsvariablen zueinander.

Data Visualization Paket

Das Visualization Paket ist ein Zusatzpaket für erweiterte Visualisierungsmöglichkeiten vor allem für mehrdimensionale Darstellungen. Das Paket beinhaltet neben sieben weiteren Darstellungsarten weitere spezifischere Funktionalitäten zur Auswertung und Weiterverarbeitung von Daten.

(2D) Histogramm Diese Visualisierung bietet viele Ergänzungen und Verbesserungen zum Standardhistogramm an. Somit können neben Einschränkungen und Zielvorgaben auch Histogramme in Abhängigkeit von zwei Variablen, also im dreidimensionalen Raum dargestellt werden.

Glyph/Star Glyph Plot Hiermit können multidimensionale Daten dargestellt, aufbereitet und verarbeitet werden. Normale Glyph Plots werden als Würfel dargestellt. Mit den drei Dimensionen (x-, y- und z-Position), Größe, Farbe, Orientierung und Transparenz können so bis zu 7-dimensionale Daten dargestellt werden. Der Star Glyph Plot stellt ein Datum als 4-armiges Koordinatensystem dar und kann so über die Länge der Koordinatenarme

eine weitere achte Dimension visualisieren.

Parallel Coordinates Diese Visualisierungsform reiht Daten mit unterschiedlichen Achsen parallel zueinander auf. Diese Achsen sind die verschiedenen zu untersuchenden Variablen. Die Verbindung zweier Variablen wird durch eine Hilfslinie dargestellt. Der Schnittpunkt mehrerer solcher Hilfslinien oder Schnittpunkte mit den Achsen geben dann Auskunft über Korrelationen.

Scatter/Scatter Matrix Scatter und Scatter Matrix bieten ergänzende und erweiterte Funktionen zur Standard Visualisierung an. Hier kann ein einzelner Scatter Plot im Detail konfiguriert und betrachtet werden und die Scatter Matrix (Zusammenfassung mehrerer Scatter Plots in einer Matrix) angepasst werden.

Für die RCE-Software sind in diesem frühen Stadium neben den allgemeinen Darstellungsformen (Tabellenansicht, Graphen und Diagramme) vor allem die Contour-Plots und der Prediction Profiler von Interesse (s. Anforderungsanalyse, Anforderungen der Entwickler).

Die Darstellung des Contour Plots in ModelCenter ist in Abbildung 6 zu finden. Im unteren Teil befindet sich der Data Explorer mit Contour Plot Ansichten. Für den Contour Plot gibt es unterschiedliche Konfigurationsmöglichkeiten abhängig von den jeweiligen darzustellenden Variablen, unter anderem die Darstellung als Kurvenschar oder als Ebene im kartesischen Koordinatensystem.

Im oberen Teil von Abbildung 6 befindet sich auf der rechten Seite zunächst die allgemeinen Einstellungen, die durch die erweiterten Einstellungen (links) ergänzt werden. Damit können Aussehen, Verhalten und unterschiedliche Betrachtungsweisen für die Contour Plot-Visualisierung realisiert werden und so die gewonnenen Daten ausgewertet und für die Weiterverarbeitung aufbereitet werden.

Mit dem Prediction Profiler können Input Variablen von Komponenten verändert werden um die zu erwartenden Ausgangsvariablen abzuschätzen. Der Prediction Profiler besitzt eine graphische Auswertung der Variablen in Form von zweidimensionalen Graphen oder Contour Plots. Zudem können über die Prediction Profiler-Werkzeugleiste die Eingangs- und Ausgangsvariablen angepasst werden (Definitionsbereiche der Eingangsvariablen, Farben der Graphen/Punkte, Wertebereiche der Ausgangsvariablen, Darstellungsform).

¹⁴vgl. Phoenix Integration (1999-2011): PHX ModelCenter Hilfe. Stand: 23. Juli 2012

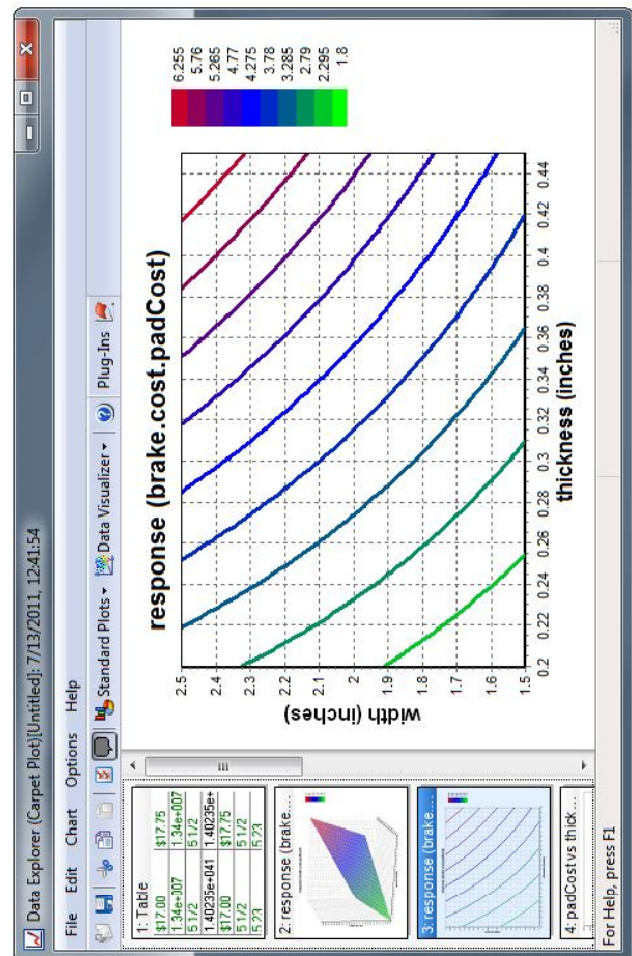
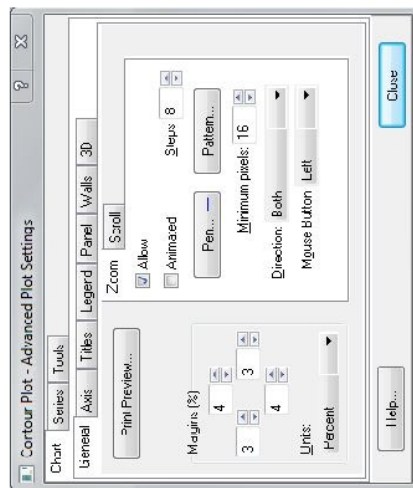
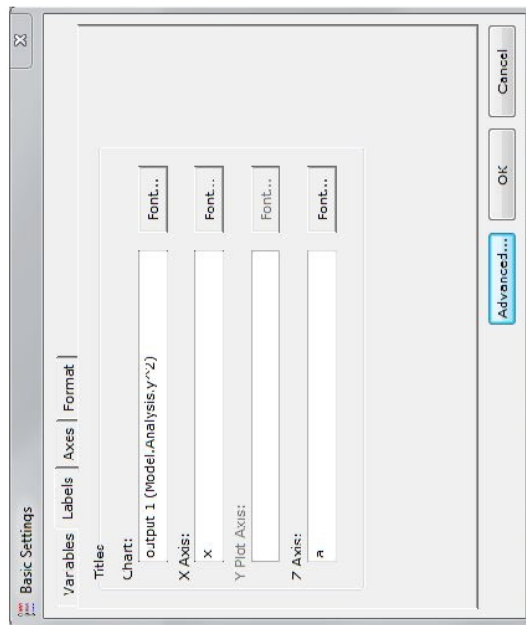


Abbildung 6: Contour Plot Ansicht in PHX ModelCenter ¹⁴

Zusätzliche Visualisierungen und Visualisierungsfunktionalitäten, angepasst an jeweilige Anwendungsfälle werden mit der Weiterentwicklung der RCE-Software einen größer werdenden Stellenwert einnehmen. Deshalb wird die RCE-Visualisierung an unterschiedlichen Stellen erweiterbar sein.

2.5 Visualisierung von wissenschaftlichen Daten in anderen Software-Produkten

Neben der RCE-Software und PHX ModelCenter ist das Thema Visualisierung von Daten und Datenmengen in vielen weiteren Softwareprodukten von zentraler Bedeutung. Im Folgenden wird eine Auswahl der relevanten Vorgehensweisen, Funktionalitäten und Konzepte vorgestellt, die interessante Aspekte für die Entwicklung der RCE-Visualisierungskomponente eröffnen und zudem eine weitere Perspektive in die Konzeption miteinbringen.

Die meisten Softwareprodukte beinhalten Visualisierungen. Zunächst wird hier eine Auswahl der vorbildlichen und erfolgreichen Produkte getroffen. Später werden dann einzelne Funktionen hervorgehoben, die für die RCE-Software interessant sind.

- Microsoft Excel
- Microsoft Pivot (Microsoft Live Labs)
- Mathematica (Wolfram Research)
- Funktionenplotter mit mathematischen Funktionen: GnuPlot, Euklid DynaGeo, GeoGebra, MatheGrafix

Anhand dieser Softwarebeispiele können unterschiedliche Konzepte beobachtet werden.

Ein umfangreiches Visualisierungskonzept bietet die Tabellenkalkulationssoftware Microsoft Excel. Grundlegend ist hierbei die Tabelle, die die zu visualisierenden Daten enthält (ähnlich PHX ModelCenter und RCE). Anhand dieser Tabelle können dann unterschiedlichste grafisch ansprechende Visualisierungsformen generiert werden, wie zum Beispiel Säulendiagramme, Kurven, Kuchendiagramme, Netz- und Punktdiagramme. Diese werden in Form eines Grafikelementes in die Arbeitsoberfläche übernommen, können konfiguriert und exportiert werden. Außerdem können in Excel Daten aus den unterschiedlichsten Quellen verarbeitet werden, wie zum Beispiel Daten aus XML-Dateien und Datenbanken.

Ein weiteres umfangreiches Softwareprodukt ist die Mathematica-Software von Wolfram Re-

search. Neben den Standardvisualisierungsmethoden der Funktionenplotter und deren mathematischen Funktionalitäten, besitzt Mathematica zusätzlich ein umfangreiches Konzept zur Wissensfindung in den verschiedenen mathematischen Bereichen. Mit Mathematica kann innerhalb eines einzigen Tools einfach und schnell technische und physikalische Funktionen bebildert und geordnet abgerufen werden, numerische Berechnungen durchgeführt und Daten analysiert werden.¹⁵ Die Darstellung in Form von Text-, Grafik-, Visualisierungs-, Analyse- und Videoelementen lässt so eine Abdeckung von vielen Themen eines Kernbereichs zu.

Ein weiteres interessantes Konzept bietet der PivotViewer von Microsoft. Dieser gliedert eine Menge von Daten in Containern. Diese Daten werden dann containerweise dargestellt. Ein einzelnes Datum wird als Minigrafik dargestellt. Daten mit gleichen Metadaten können dann in Form dieser Minigraphiken unterschiedliche präsentiert werden. Der PivotViewer beherzigt das in Kapitel 2.1 vorgestellte Visual Information Seeking Mantra in einer sehr ansprechenden Weise. Obwohl der PivotViewer für die Suche im WWW angedacht ist, bietet das Grundkonzept doch einen interessanten Punkt für ein Visualisierungskonzept für die RCE-Software.¹⁶

¹⁵vgl. Wolfram Research (2012): Mathematica Software.

¹⁶vgl. Microsoft Silverlight (2012): PivotViewer

3 Anforderungsanalyse

Die in diesem Kapitel behandelte Anforderungsanalyse bildet einen wichtigen Teil dieser Bachelorarbeit. Dazu wird kurz das verwendete Vorgehensmodell erläutert, die Sammlung und Strukturierung der einzelnen User Stories aufgezeigt und daraus die Anforderungen an die RCE-Visualisierung definiert.

3.1 Vorgehensweise

Zur Erfassung der unterschiedlichen Meinungen und Erfahrungen der Entwickler und Benutzer werden User Stories benutzt. Aus diesen werden dann die Anforderungen an ein Konzept erarbeitet.

User Stories dienen in der agilen Softwareentwicklung zur Formulierung von Anforderungen aus Sicht des Kunden. Diese sind bewusst in Alltagssprache gehalten. Zu jeder User Story gehört die Rolle des Autors, das Ziel der User Story und der daraus entstehende Nutzen. Im Gegensatz zur Anforderungsbeschreibung mit Use Cases sind User Stories wesentlich kürzer und weniger statisch. Für die Entwicklung eines Visualisierungskonzeptes gibt es viele Anforderungen aus den unterschiedlichen Bereichen, die sich mit dem starren Use-Case-Modell nur schwierig und mit großem Aufwand zusammenfassen und beschreiben lassen. Zudem wird zunächst ein Idealkonzept erarbeitet (s. hierzu Kapitel 4) auf dem dann ein für RCE praktikables Konzept aufbaut. Dieses sollte zum einen in der Entwicklungsphase hinzukommende Änderungen berücksichtigen und außerdem neue Anforderungen, die mit Neuerungen der RCE-Software entstehen, umgehen können.

Im Rahmen dieser Bachelorarbeit erfolgt die Beschreibung der Anforderungen deshalb mittels User Stories.

Um ein gutes Mittelmaß zwischen Detailgrad der einzelnen und Anzahl der gesamten User Stories zu erhalten, werden die einzelnen User Stories in verschiedene Bereiche und Themen eingeteilt und übergeordnete User Stories (Epics) verfasst. Dies dient zum einen der Struktur und Übersichtlichkeit. Zum anderen können neu hinzu gekommene Anforderungen eingeordnet und der Aufwand leichter abgeschätzt werden.¹⁷

Durch regelmäßige Rücksprache mit dem RCE-Entwicklerteam sind viele Anforderungen an

¹⁷vgl. Wirdemann (2011): SCRUM mit User Stories. S. 50ff, S. 104ff

ein Visualisierungssystem bereits abgedeckt. Diese betreffen in erster Linie nichtfunktionale und technische Aspekte der Visualisierung. Die Motivation der Verbesserung der Visualisierung ist aus bereits eingegangenen Anforderungen entstanden.

Um ein weites Spektrum an Perspektiven und Meinungen berücksichtigen zu können, sind darauf aufbauend die Hauptanwender der RCE-Software nach deren aktuellem Gebrauch von Visualisierungswerkzeugen innerhalb und außerhalb der Software befragt worden. Deren Meinungen und Präferenzen zu einer RCE-Visualisierungskomponente sind maßgeblich für ein Visualisierungskonzept. Mittels eines Fragebogens sind die Benutzer zu ihrem aktuellen Vorgehen in RCE-Visualisierungsfragen, zu vordergründigen Funktionalitäten, der Integration in die RCE-Software und ihren bevorzugten Visualisierungsarten befragt worden.¹⁸

Der Anwenderbefragung liegt die Prämisse zugrunde, dass bei der Anforderungserhebung etwaige Umsetzungsmöglichkeiten und technische Restriktionen ausgeblendet werden und zunächst der Idealzustand betrachtet wird. Der Fragebogen dient dabei als Einstieg und Leitfaden.

Zu den Anforderungen aus Entwickler- und Benutzerinterviews stellt abschließend die Analyse anderer Softwareprodukte und -systeme eine Anforderungs- und Informationsquelle dar.

3.2 Einsatzszenarios und angestrebte Verbesserungen

Die einzelnen Schritte bei der Visualisierung von Workflow-Daten lassen sich in einzelne Bereiche unterteilen. Diese Bereiche bilden die Epics (Überbegriffe) der User Stories. Die RCE-Visualisierung ist unterteilt in:

- Erstellen eines Workflows/einer Simulation/einer Studie
- Komponenten- und workflowspezifische Visualisierung
- Visualisierung zur Laufzeit
- Visualisierung zur nachträglichen Auswertung der Daten

Das Erstellen eines Workflows in RCE wird zurzeit über den Workflow-Editor (s. Kapitel 'Übersicht über die RCE-Software') realisiert und muss nicht neu konzipiert werden. Da dieser jedoch ebenfalls zur Visualisierung in RCE gehört, wird er hier aufgeführt und Möglichkeiten der Verbesserung aufgezeigt. Bei der komponentenspezifischen Visualisierung werden

¹⁸Fragebogen und Zusammenfassung der Antworten siehe Anhang A, B

Eingangs- und Ausgangsdaten einer einzelnen Komponente eines Workflows (Optimierer, Parameterstudie,...) betrachtet und diese Daten dargestellt. Die workflowspezifische Visualisierung visualisiert im Gegensatz dazu Input und Output eines gesamten Workflows, bzw einer gesamten Studie. Benutzer der RCE-Software wollen die Daten sowohl zur Laufzeit der Workflows, als auch zur Nachbearbeitung visualisieren können. In beiden Bereichen ergeben sich dabei spezielle Anforderungen.

Diese Epics für die einzelnen User Stories werden nachfolgend im Detail diskutiert.

Zunächst ergeben sich zur Verbesserung des bestehenden Workflow-Editors folgende Anforderungen (s. Tabelle 1).

Darstellung des Workflows

Der Benutzer

- ... möchte im Workflow-Editor Beschreibungen einfügen.
- ... möchte Notizen zum Workflow hinzufügen.
- ... möchte Labels an die Workflow-Elemente anheften.
- ... möchte mehrere Komponenten zusammenschließen und gruppieren
- ... möchte die Visualisierungskomponenten aus dem Workflow-Editor heraus öffnen.

Tabelle 1: User Stories für die Darstellung eines Workflows

Der Workflow-Editor ist unter Benutzung des Eclipse Frameworks GEF (Graphical Editing Framework) implementiert. Dieses Framework liefert die notwendige Technik um in der Eclipse RCP-Oberfläche grafische Editoren und Views zu erstellen. Es handelt sich dabei um ein MVC-Framework (Model View Controler).¹⁹

Die eigentliche Visualisierung in der RCE-Software betrifft die Ausführung des Workflows und die damit entstehenden Daten. Die Anforderungen an die Visualisierungskomponente betreffen die anfallenden Daten der einzelnen Komponenten und der gesamten Studie. Die Anforderungen an ein Visualisierungskonzept sind in Tabelle 2 als User Stories dargestellt.

¹⁹vgl: Eclipse Foundation (2012b): GEF(MVC)

Visualisierung von komponenten- und workflowspezifischen Daten

Der Benutzer

- ... möchte über eine Schaltfläche die Visualisierung zum aktiven Workflow öffnen.
- ... möchte im Workflow-Editor eine Komponente auswählen und die dazugehörigen Daten visualisieren.
- ... möchte mehrere Komponenten zugleich visualisieren.
- ... möchte die Daten zur Laufzeit und zur Nachbearbeitung visualisieren.
- ... möchte aus unterschiedlichen Visualisierungsformen wählen können.
- ... möchte die Visualisierung aus dem Editor heraus öffnen.
- ... möchte sowohl Ein- und Ausgänge einer Komponente, als auch die eines gesamten Workflows visualisieren können.

Tabelle 2: User Stories für die Visualisierung von komponenten- und workflowspezifischen Daten

In ModelCenter werden nur Input und Output von einzelnen Komponenten visualisiert. Um Daten von mehreren Komponenten oder eines gesamten Workflows zu visualisieren, werden die Komponenten in ModelCenter in Assemblies gruppiert und so nach außen als einzelne Komponente gekapselt. In der RCE-Software ist dies ebenfalls angedacht und eine getrennte Betrachtung von workflow- und komponentenspezifischen Daten nicht nötig. Der Vollständigkeit halber sind beide Fälle hier aufgelistet (letzter Punkt der Tabelle 2).

Die anfallenden Daten werden zusätzlich zu der Unterteilung auf Komponente und Workflow zu verschiedenen Zeitpunkten betrachtet. Die Ausführung eines Workflows kann in bestimmten Fällen Zeit in Anspruch nehmen, während der man den Verlauf der Simulation überwachen will. Deswegen sollen in der RCE-Visualisierungskomponente die Daten auch zur Laufzeit angezeigt werden. Um eine Überwachung der Daten zur Laufzeit des Workflows (bzw. der Studie, der Simulation) zu ermöglichen, gibt es diesbezüglich Anforderungen, die unter anderem in den User Stories in Tabelle 3 aufgelistet sind.

Darstellung der Daten zur Laufzeit, Nachbearbeitung und Auswertung

Der Benutzer

- ... möchte, dass die Visualisierung beim Ausführen des Workflows geöffnet wird.

- ... möchte ein dynamisches asynchrones Updaten der Datentabelle und der geöffneten Visualisierung zur Laufzeit.
 - ... möchte Angaben zur Anzahl der Aufrufe und benötigter Zeit einer Komponente pro Studie und pro individuellem Aufruf erhalten.
 - ... möchte bei der Optimierer-Komponente die Zielfunktion pro Optimierungsiteration sehen.
 - ... möchte zu mehreren Studien Visualisierungen öffnen können.
 - ... möchte gängige mathematische Funktionen benutzen (Intervalle, Interpolation, Maximum/Minimum,...).
 - ... möchte benutzerspezifische Einstellungen an der Darstellung vornehmen.
 - ... möchte visualisierte Daten in unterschiedlichen Formaten exportieren.
 - ... möchte bei der Optimierer-Komponente die Zielfunktion pro Optimierungsiteration sehen.
 - ... möchte statistische Werkzeuge zur Auswertung und Bearbeitung der Daten
-

Tabelle 3: User Stories für die Darstellung der Daten zur Laufzeit, Nachbearbeitung und Auswertung

Der Hauptanwendungsfall für die RCE-Visualisierung liegt in der Nachbearbeitung und Auswertung der anfallenden Daten. Diese erfolgt nach Beendigung der Workflow-Ausführung. In Tabelle 3 sind die User Stories für diesen Hauptanwendungsfall zu finden.

3.3 Funktionale Anforderungen

Betrachtet man nun das gesamte RCE-Visualisierungskonzept, so bleiben neben den Anforderungen die aufgeführten User Stories noch einige komplexere Anwendungsfälle, die in einem Konzept wünschenswert wären. Diese werden in diesem Kapitel vorgestellt und erläutert.

Der Prediction Profiler soll ein umfangreiches Werkzeug zum Abschätzen von Ausgangswerten /-variablen für ein gegebenes Problem darstellen. Die Ausgangswerte sollen für unterschiedliche Kombinationen von Eingangsvariablen dargestellt gefunden werden. Dieses Werkzeug ist vor allem für ein manuelles Durchsuchen von Werten und Variablen darstellen um so zum Beispiel Potential für einen Optimierungsalgorithmus herauszufinden oder um nach einem Optimierungsprozess händisch Anpassungen vorzunehmen und deren Resultat zu visualisie-

Funktionale Anforderungen an die RCE-Visualisierung

Der Benutzer

- ... möchte anhand von Eingangsvariablen die Ausgangswerte abschätzen ('Prediction Profiler').
- ... möchte die Datenflüsse zwischen den Komponenten visualisieren ('Provenance').
- ... möchte die Visualisierungskomponente über eine Schnittstelle in eigene Programme einfügen.
- ... möchte Graphen unterschiedlicher Komponenten und Visualisierungsformen miteinander vergleichen können (in getrennten Fenstern, in einem gemeinsamen Koordinatensystem).
- ... möchte visualisierte Daten als Vektorgrafik exportieren.
- ... möchte Daten einer Studie in ASCII-Dokument exportieren (*.csv).
- ... möchte externe Daten in RCE importieren um diese mit der RCE-Visualisierungskomponente darzustellen.

Tabelle 4: User Stories für die Nachbearbeitung und Auswertung der Simulation und der Daten

ren.

Um die RCE-Visualisierung in eigene Programme (z.B. in ein eigenes Python-Skript) einzufügen, benötigt die Visualisierungskomponente eine Schnittstelle, die Funktionalitäten bereitstellt und die Integration in andere Anwendungen ermöglicht.

Bei CSV (= Comma Separated Values) handelt es sich um ein Dateiformat, das den Aufbau einer Textdatei beschreibt. Dadurch können Daten zwischen unterschiedlichen Anwendungen über Export- und Importfunktionen ausgetauscht werden. CSV-Files können unterschiedlich definiert sein.²⁰

3.4 Nichtfunktionale Anforderungen

In diesem Abschnitt werden die nichtfunktionalen Anforderungen an die Integration der Visualisierung von Daten in RCE dargelegt. Diese legen neben allgemeinen Anforderungen an

²⁰vgl. IETF (2005): RFC 4180: Common Format and MIME Type for Comma Separated Values (CSV) Files.

Stand: August 2012

die RCE-Entwicklung auch fest, welche Eigenschaften die RCE-Visualisierung in Bezug auf die Integration und Handhabung haben sollte.

Java, Eclipse RCP, RCE-Standards: Die Implementierung sollte in der Programmiersprache Java stattfinden. Die Eclipse RCP als Basissoftware und Implementation als Eclipse-Plugin ist beizubehalten. Die allgemeinen RCE-Standards und -Konventionen sind einzuhalten (z.B. Architekturrichtlinien, Code-Style-Festlegungen, Programmier- und Dokumentationsrichtlinien).

Visual Information Seeking Mantra: Um eine gute Usability und so die Zweckmäßigkeit der RCE-Visualisierung garantieren zu können, ist eine grobe Einhaltung des Visual Information Seeking Mantras (s. Kapitel 2.1) anzustreben.

Externe Bibliotheken: Da es sich bei der RCE-Software um ein Open-Source Projekt handelt, müssen auch die benutzten Bibliotheken unter einer entsprechenden Lizenz verfügbar sein.

Integration in die RCE-Software: Aus den Antworten der Entwickler auf den Fragebogen geht hervor, dass eine RCE-Visualisierung außerhalb der Anwendung (als externes Programm) nicht erstrebenswert ist. Somit sollte die Integration innerhalb der RCE-Software stattfinden, bestenfalls als eigenständiges separates Visualisierungsfenster.

Zeitpunkt der Visualisierung: Die Daten sollen sowohl während der Workflow-Laufzeit, als auch nach ausgeführter Simulation (post-process) visualisiert werden können.

Erweiterbarkeit: Die RCE-Software wird stetig weiter entwickelt und neue Anwendungsbereiche und Benutzerkreise akquiriert. Deshalb soll die Visualisierungskomponente leicht erweiterbar sein für neue Anwendungsfälle.

Somit ergeben sich eine Reihe von Anforderungen, die zur Qualitätsmessung des Konzeptes und der Visualisierung selbst herangezogen werden können (s. Schlusskapitel). Hiermit sind wichtige allgemeine Präferenzen der RCE-Benutzer (und -Entwickler) abgedeckt.

3.5 Rahmenbedingungen

Die Entwicklung des RCE-Visualisierungskonzept (s. Kapitel 'Entwurf eines Visualisierungskonzeptes) betrachtet zunächst keine Einschränkungen, die sich beispielsweise aus der Eclipse

RCP ergeben. Es stellt sich hier nun die Frage, ob die RCE-Visualisierung mithilfe der gängigen RCP-Mechanismen integriert wird, oder ob eine neue Softwarekomponente ohne optischen Bezug vorteilhafter ist. Hierfür werden in diesem Kapitel Rahmenbedingungen erläutert.

Die Rahmenbedingungen für die RCE-Visualisierungen ergeben sich hauptsächlich aus den Gegebenheiten, die die Eclipse RCP als Basisplattform mit sich bringt.

Im Kapitel 2.2 'Übersicht über die RCE-Software' sind bereits die Grundzüge und die Arbeitsweise der Eclipse RCP dargestellt. Zusätze zur Plattform sind in der RCP über sogenannte Extension Points möglich. Dabei handelt es sich um vordefinierte Schnittstellen, an denen Zusätze zur Software angebracht werden können. Hierbei gibt es von der Eclipse RCP und selbst definierte Extension Points. Für die RCP-Oberfläche existieren fünf unterschiedliche Möglichkeiten, Elemente zur Eclipse RCP hinzuzufügen: Editoren, Views, Dialoge, Wizards und Aktionen/Kommandos.

Editoren werden genutzt um bestimmte Objekttypen, wie Java-Klassen, Text-Dateien, Bilder und Ähnliches zu öffnen. Views beschreiben einzelne Objekte näher und Dialoge ermöglichen die Interaktion des Nutzers. Wizards werden genutzt, um einen bestimmten Ablauf zu strukturieren und zu vereinfachen. Aktionen und Kommandos sind Funktionalitäten, die an bestehende oder selbst definierte Editoren und Views, oder aber an die Eclipse-RCP selbst angehängt werden. Aktionen sind dabei fest mit dem jeweiligen Oberflächenelement verbunden, wohingegen Kommandos ausgelagert sind und über Handler aufgerufen werden.

Als Rahmenbedingung für die Umsetzung der RCE-Visualisierung ergibt sich somit eine Integration der Visualisierung über die vorgesehenen Extension Points der Eclipse RCP.

4 Entwurf eines Visualisierungskonzeptes

Anhand der zusammengetragenen Anforderungen aus dem vorangegangenen Kapitel wird hier nun das erarbeitete Visualisierungskonzept vorgestellt. Dieses versucht in erster Linie die Anforderungen an die Oberfläche umzusetzen und geht dann auf die Schnittstelle zwischen der bestehenden Software und der Visualisierung ein. Abschließend wird in diesem Kapitel der Hintergrund des praktischen Teils dieser Bachelorarbeit erläutert.

4.1 Abläufe und Aufmachung der Visualisierung

Für die Konzeptentwicklung werden zunächst technische oder wirtschaftliche Einschränkungen außen vor gelassen und das Hauptaugenmerk auf eine gut benutzbare Datenvisualisierung konzentriert. Im Detail werden hierfür neben Abläufen in der Oberfläche, auch einzelne notwendige Elemente der Visualisierung vorgestellt und Design, Aufbau, Layout und Struktur betrachtet.

4.1.1 Ablauf der Visualisierung von Daten in der Oberfläche

Die Visualisierung von Daten in RCE betrachtet prinzipiell einen einzigen zeitlichen Ablauf von Arbeitsschritten in der RCE-Software. Dieser Ablauf ist in der Abbildung 7 vereinfacht schematisch zu sehen.

Zunächst wird im Workflow-Editor ein Workflow mit einzelnen Komponenten erstellt. Wird der Workflow ausgeführt, öffnet sich automatisch die Datenvisualisierung. Diese zeigt zunächst bei noch nicht abgeschlossener Ausführung des Workflows die Laufzeitvisualisierung an. Sobald der Workflow durchlaufen ist, wird die Visualisierung zur Nachbearbeitung und Auswertung der simulierten Daten bereitgestellt. Die Visualisierung zur Laufzeit und die Visualisierung zur Nachbearbeitung und Auswertung unterscheiden sich im Wesentlichen nur in ihrem Funktionsumfang. Zur Laufzeit benötigt der Nutzer eine übersichtliche Darstellung der Daten. Eine mathematische Auswertung von einzelnen Werten ist dabei hintergründig. Zur Laufzeit ist ein komplexes mathematisches Bearbeiten und Betrachten der einzelnen Daten überflüssig. Zu diesem Zeitpunkt wird der Fortschritt der Workflow-Ausführung und der Simulation überwacht, um mögliche Fehler frühzeitig zu detektieren und zu beheben. Zur Nachbearbeitung sollen dann komplexere Funktionalitäten und Bearbeitungsmöglich-

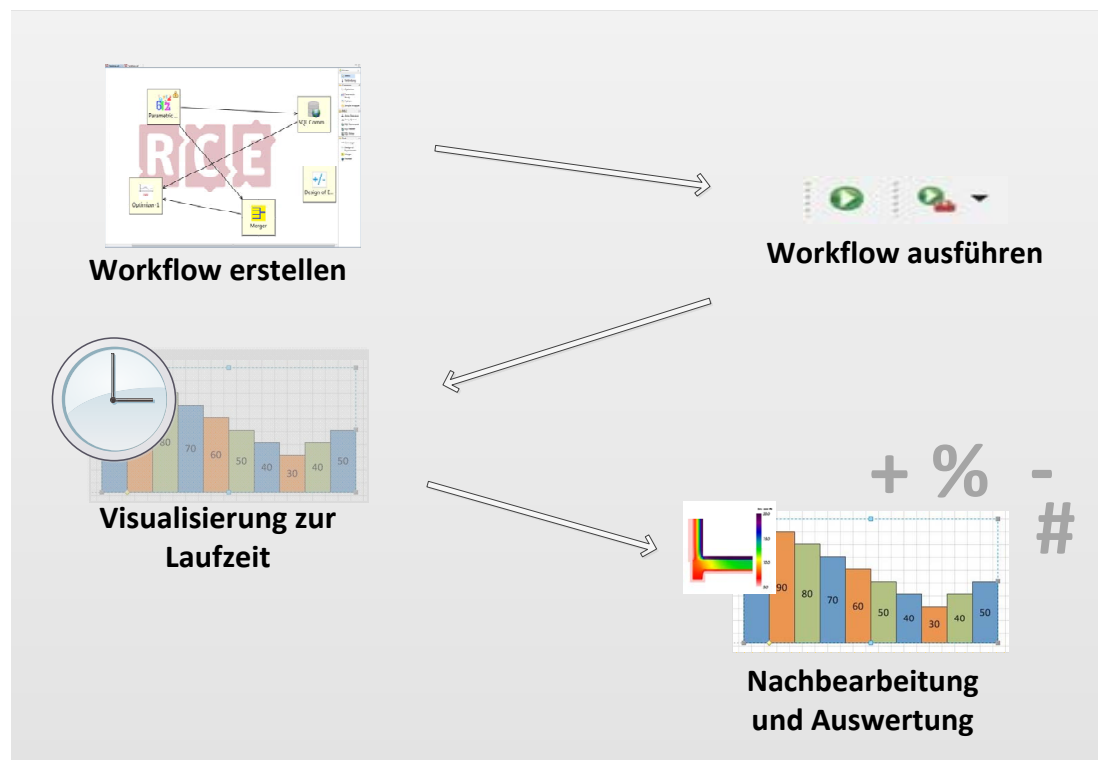


Abbildung 7: Standardablauf einer Workflow-Visualisierung

keiten zur Verfügung stehen. Die Daten sollen hier umfassend ausgewertet, die Darstellung angepasst und dann zur weiteren Verarbeitung (zum Beispiel in Form von Berichten und wissenschaftlicher Dokumentationen der Simulationen) bereitgestellt werden.

Welche Komponente bei der Ausführung des Workflows visualisiert wird, richtet sich nach der Auswahl im Workflow-Editor. Ist eine einzelne Komponente ausgewählt, wird der jeweilige In- und Output der gewählten Komponente dargestellt. Ist keine Komponente ausgewählt, werden die Ein- und Ausgänge des gesamten Workflows betrachtet. Hierbei befinden sich alle Komponenten des Workflows in einem Container, der für die Visualisierung wiederum Komponentencharakter besitzt (vgl. hierzu Assembly im Kapitel 2.4.1 'Abgrenzung von RCE-Software und PHX ModelCenter').

Neben diesem Hauptanwendungsfall gibt es noch einige zweitrangigere Abläufe, die zur Visualisierung von Daten mit der RCE-Visualisierung führen. Diese sind im Einzelnen:

- Visualisierung wird außerhalb des Workflows geöffnet → Zu visualisierende Workflows und Komponenten ausgewählt → Daten werden wie gewohnt dargestellt

- Workflow wird ohne Editor ausgeführt → Visualisierung des gesamten Workflows
- Workflow wird ohne RCE-Client ausgeführt (headless) → Keine Visualisierung
- Visualisierung wird geöffnet → Externe Daten werden importiert → Daten werden in RCE dargestellt

Diese alternativen Abläufe betrachten zunächst die unterschiedlichen Möglichkeiten einen Workflow auszuführen (ohne Editor über Menüeintrag oder per Rechtsklick im Project Explorer). Solange die RCE-Oberfläche genutzt wird, öffnet sich dabei die Visualisierung. Wird ein Workflow ohne Benutzung der RCE-Oberfläche ausgeführt, so wird auch die Visualisierung nicht gestartet (Headless-Modus ist für zukünftige RCE-Versionen angedacht). Zudem sollen externe Daten in RCE mit der Visualisierung dargestellt werden können. Hierzu soll es in der Visualisierung die Möglichkeit geben.

4.1.2 Layout und Struktur der Visualisierung

Im Grundlagenkapitel wird das Information Seeking Mantra von Ben Shneiderman vorgestellt. Dabei wird ein generelles, abstraktes Vorgehen bei der Informationssuche und Auswertung von großen Datenmengen durch eine Abfolge von Punkten/Schritten beschrieben. Dieses Prinzip lässt sich auf die Konzeption der RCE-Visualisierung anwenden.

Die einzelnen Elemente des Visualisierungskonzeptes werden dabei an einen bestimmten Schritt des Information Seeking Mantras angepasst und diesem zugeordnet. Das Visualisierungskonzept beinhaltet folgende Oberflächen-Elemente:

Datentabelle: Die bereits vorhandene Darstellung der Daten in Tabellenform ist zwingend erforderlich. Die Tabelle, die alle Werte beinhaltet, soll am unteren Rand permanent sichtbar sein, um Überblick und Referenzen schaffen zu können.

Diagramm-/Kurvenbereich: Der Diagramm- und Kurvenbereich ist das Kernstück der Visualisierungsoberfläche und beinhaltet die eigentliche Darstellung und Interpretation der Daten. In diesem Bereich werden die Daten mit der ausgewählten Darstellungsart in Form von Diagrammen, Graphen, Kurven und Tabellen gezeigt.

Komponenten: Für einige Komponenten sind neben den Ein- und Ausgängen zudem weitere Informationen wie Variablennamen, Gleichungen und Informationen zu Optimierungsparametern erforderlich. Diese werden in das Diagramm- und Kurvenbereichselement

integriert.

Auswahl an Visualisierungsformen: Zur Auswahl unterschiedlicher Visualisierungstypen und Anwendungsbereiche gibt es ein Element, das die einzelnen Typen zur Verfügung stellt.

Historie/ Zuletzt verwendet: Um die Arbeit mit den visualisierten Daten zu verbessern, gibt es einen Bereich neben der Auswahl der Visualisierungen, der die zuletzt genutzten Typen und Werkzeuge zeigt. Dies dient der Fehlererkennung und Nachvollziehbarkeit von Änderungen.

Menüleiste: Die Menüleiste ist für allgemeine Funktionalitäten angedacht. Hier finden sich zusätzlich auch Menüs mit den vorhandenen Werkzeugen und Optionen.

Werkzeugleiste: Die Werkzeugleiste dient zur leichteren und schnelleren Arbeit mit der Visualisierung und beinhaltet beispielsweise einen Eintrag zu den mathematischen Funktionalitäten.

Inhalte der einzelnen Elemente werden in einem späteren Kapitel (Kapitel 4.2) näher erläutert. Die gesamte Aufmachung der Visualisierung soll optisch zur RCE-Software passen. Das Layout ist daher klassisch und praktisch gestaltet, in Anlehnung an RCE und ModelCenter, mit wenig Style-Elementen und aufwändigem Design.

Das gesamte Konzept mit den einzelnen genannten Elementen ist in Abbildung 8 zu sehen. Im Mittelpunkt befindet sich der Diagramm- und Kurvenbereich, ergänzt durch nähere Informationen. Am unteren Rand ist die Datentabelle. Diese ist bereits wie die Leisten und Auswahlfelder bereits beim Öffnen des Fensters zu sehen. Zu Beginn der Datenauswertung gibt die Tabelle den ersten Überblick über die Gesamtheit der Daten. Sie ist fester Bestandteil und gewährleistet, dass wichtige Bereiche nicht ausgelassen werden und Relationen global erkannt werden können. Daten, Variablen und Werte können so jederzeit eingeordnet werden, ohne dass die eigentliche Detailarbeit unterbrochen werden muss.

Am rechten Rand der Abbildung 8 befindet sich das Auswahlpaneel für die einzelnen Visualisierungsarten und Anwendungsbereiche. Darunter stehen die zuletzt geöffneten Typen und die jeweils durchgeführten Aktionen als Liste bereit. Am linken Rand ist die Werkzeugleiste zu sehen. Diese beinhaltet Schaltflächen um die einzelnen Konfigurations- und Funktionsdialoge zu öffnen (vgl. Kapitel 4.2). Die Menüleiste am oberen Rand dient dazu, allgemeine Funktionen des Visualisierungsfenster bereitzustellen, wie Schließen und Minimieren/Maxi-

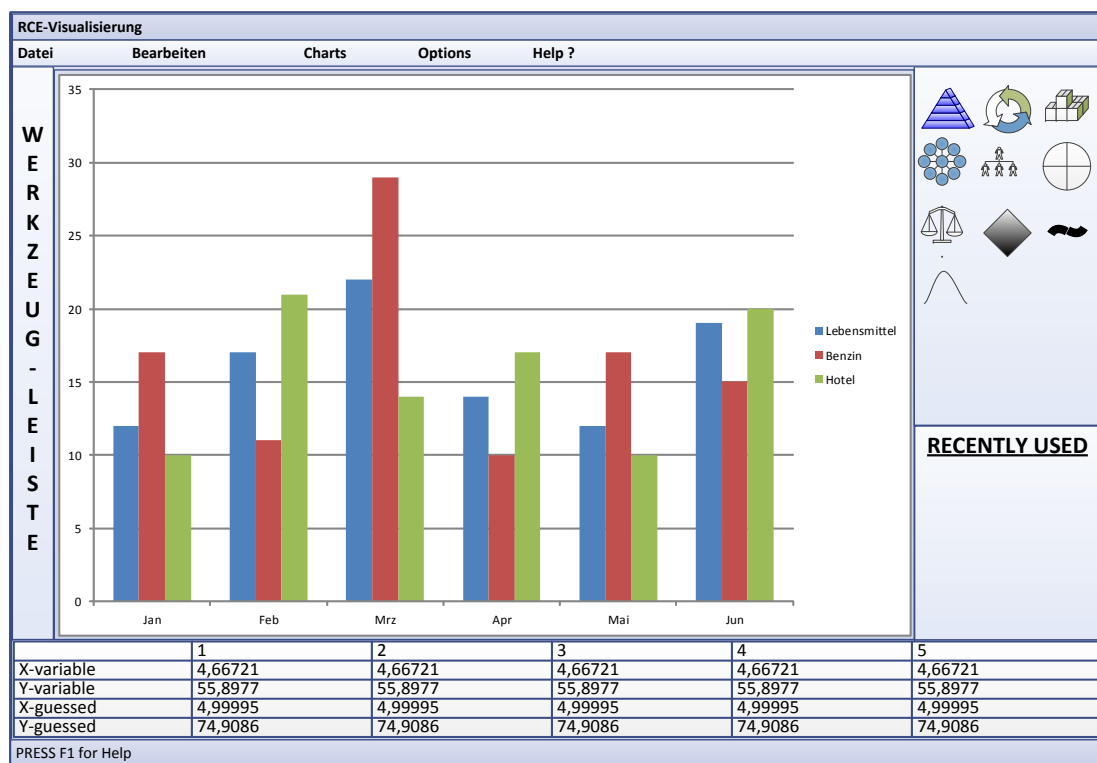


Abbildung 8: Konzept der RCE-Visualisierung mit einzelnen Elementen

mieren. Zudem gibt es in den einzelnen Menüs zusätzlich die Möglichkeit, die Funktionalitäten der Werkzeugleiste und Konfigurationsdialoge zu öffnen. Die Menüleiste ist aus Konsistenzgründen vorhanden. Die einzelnen Aktionen werden vor allem aus den einzelnen Elementen heraus gestartet.

Die einzelnen Elementen können jeweils zu den Punkten des Information Seeking Mantras zugeordnet werden (vgl. Kapitel 2.1). Die Tabelle bildet zusammen mit dem Workflow-Editor den Überblick (Overview), mit der Auswahl von Visualisierungsart und Werkzeugen können einzelne Punkte detaillierter dargestellt und gefiltert werden (Zoom Filter). Graphen, Punkte, Variablen und einzelne Werte können bei Bedarf abgerufen werden und mit den unterschiedlichen Funktionalitäten in Relation gesetzt werden (Details-on-demand, Relate). Durch die Liste der zuletzt genutzten Elemente wird eine Historie bereitgestellt und letztendlich können die gewonnenen Erkenntnisse in Form von Grafiken exportiert und festgehalten werden.

4.2 Funktion und Inhalt der einzelnen Elemente der Visualisierungsoberfläche

Dieses Kapitel beschreibt die Funktionen und Möglichkeiten der Elemente der RCE-Visualisierung genauer. Hierzu werden zunächst der Inhalt der Leisten, darauf aufbauend die Funktionsbereiche und Konfigurationspaneele und im Anschluss, die Palette an Visualisierungsformen erläutert.

4.2.1 Inhalt der Werkzeug- und Menüleiste

Inhalte der Menü- und Werkzeugleiste überschneiden sich an manchen Stellen. Generell ist jedoch die Menüleiste für globale, die Visualisierung im gesamten betreffende Funktionen und Aktionen zuständig, wohingegen die Werkzeugleiste die Funktionen zur Bearbeitung und Auswertung der Daten kategorisiert und den schnellen Zugriff darauf bereitstellt. Die Menüleiste ist in vielen Applikationen und Softwareprodukten wiederzufinden und vermittelt dem Benutzer das Gefühl, die Grundstruktur bereits zu kennen. Sie unterstützt durch Konsistenz die Benutzerfreundlichkeit des Produktes.

Menüleiste

Die Menüleiste hat sechs Haupteinträge, davon sind mindestens drei aus diversen anderen Produkten bereits bekannt: Datei, Bearbeiten und Hilfe. Zusätzlich befinden sich in der RCE-Visualisierung nun noch drei weitere Einträge: Charts, Optionen und Werkzeuge.

Unter diesen Einträgen verbergen sich folgende Aktionen:

- **Datei:** Allgemeine Funktionen, wie Speicherung, Export/Import von Daten, Refresh und Fenster schließen, Liste mit zuletzt geöffneten Komponenten
- **Bearbeiten:** Aktionen Rückgängig machen/wiederherstellen, Ausschneiden/Kopieren/Einfügen, Delete, Select,...
- **Help:** Hilfeseite zur Visualisierung
- **Charts:** Liste mit den verfügbaren Visualisierungsformen
- **Options:** Aktionen, die den Diagramm- und Kurvenbereich betreffen, wie das Drucken, Exportieren als Grafik, Einheiten ändern/anzeigen/ausblenden,...

- **Werkzeuge:** Beinhaltet die Kategorien der Werkzeugleiste

Der Menüeintrag für die Werkzeuge und die Auswahl der Visualisierung ist nur der Vollständigkeit wegen und aus Konsistenzgründen aufgenommen. Die dazugehörig Funktionalität wird im Folgenden beschrieben.

Werkzeugleiste

Diese Leiste beinhaltet analog zum Menüeintrag alle Funktionen, die den Diagramm- und Kurvenbereich betreffen. Diese Funktionen sind in Kategorien unterteilt. Jede Kategorie bietet einen eigenen Satz an Funktionen, die über einen eigenen Bereich angepasst werden können. Folgende Kategorien sind zunächst vorgesehen:

- Mathematische Funktionen
- Funktionen zur statistischen Auswertung
- Anpassung von Graphen (Beschriftungen, Farben, Schriftart, ...)
- Einfügen von Textfeldern
- Export von Grafiken, Import von externen Daten
- Druckfunktion
- Einstellungen für den Arbeitsbereich (metrisches/amerikanische Einheiten)
- Prediction Profiler Funktionalität
- Vergleich von Komponenten, Iterationen und Workflows
- Design of Experiments Funktionen

Die Werkzeugleiste ist so konzipiert, dass neue Anwendungsgebiete als neue Kategorie mit einer neuen Zusammenstellung von Funktionen hinzugefügt werden können, ohne dass sich die Grundstruktur der Visualisierung ändert. Ebenso können neue Visualisierungsarten hinzukommen. Zur Erweiterung müssen dann nur die einzelnen Container-Elemente (Auswahl der Visualisierung, Werkzeugleiste) angepasst werden. Diese Modularisierung gewährleistet eine einfache Wartung und Fehlersuche und garantiert, dass die Visualisierungskomponenten anpassbar ist und auch in Zukunft benutzbar bleiben.

4.2.2 Funktionsbereich und Funktionalitäten

Die einzelnen Bereiche der Funktionskategorien beinhalten die eigentlichen Funktionen und ihre Konfiguration. Neben den Standarddialogen, wie die Druckeinstellungen, Export/Import-Konfigurationen und den Einstellungen für den Arbeitsbereich (z.B. Einstellung des Einheitensystems, Benennung von Visualisierungen, etc) sind vor allem die Mathematik- und Statistik-Kategorie und der Prediction Profiler (s. Kapitel 3 'Anforderungen'). Beispiele für Funktionen dieser Kategorien sind im Folgenden dargestellt:

Mathematik-Funktionen

- Definitions-/Wertebereiche, Intervalle
- Maxima/Minima von Funktionen
- Schnittpunkte, Nullstellen, y-Achsenabschnitte
- Integrale, Fläche unter dem Graphen
- Ableitungen, Ableitungsfunktionen
- Symmetrien, Umkehrfunktionen, Tangenten-/Normalengleichungen

Die Mathematikategorie beinhaltet Standardfunktionen der Algebra und Analysis. Die Liste der möglichen Funktionalitäten ist enorm. Eine Auswahl von Standardmöglichkeiten ist für den Anfang ausreichend und kann später erweitert werden.

Oftmals müssen die anfallenden Daten statistisch ausgewertet oder zu gegebenen Daten Algorithmen zur Verbesserung erarbeitet werden. Hierzu kann ein Repertoire an statistischen und numerischen Auswertungsmöglichkeiten sehr hilfreich sein.

Stochastische und numerische Funktionen

- Modus, Median, arithmetischer/geometrischer Mittelwert,...
- Standardabweichung, Erwartungswert, Varianz,...
- Verteilungen
- Interpolation
- Approximation
- numerische Lösungsverfahren

4.2.3 Graphen, Kurven, Tabellen, Diagramme

Hier findet sich eine Übersicht über die aktuell vorgesehenen Visualisierungsarten. Diese Liste ist ein Auszug aus den unterschiedlichen Anforderungen und ist erweiter- und anpassbar. Eine Priorisierung der Darstellungsarten ergibt sich aus der Reihenfolge.

- Koordinatensysteme mit Darstellungen unterschiedlicher Kurven, Punkten und Flächen unter dem Graphen, Ebenen und Figuren
- weitere Tabellendarstellungen
- Balken-/ Kuchendiagramme
- Histogramme
- Contour/Carpet Plots
- Scatter Plots und Interpolation
- Darstellung multidimensionaler Daten

Diese Visualisierungsformen bilden eine gute Grundlage für die Visualisierung der RCE-Daten. Weitere Darstellungen, wie die Visualisierung von Prozessen, Organigramme, Zyklus-, Radial-, Pyramiden- oder Venn-Diagramme sind nice-to-have, aber für die Grundkonzeption der RCE-Visualisierung nicht notwendig. Bei Bedarf können solche speziellen Diagramme später hinzugefügt werden.

Durch die Trennung von Visualisierungsart von den eigentlichen Funktionen entstehen weitläufige Möglichkeiten Daten, Informationen, Trends, Erwartungen und Details aus einer Simulation zu gewinnen. Die einzelnen Visualisierungen können einfach und sinnvoll an den jeweiligen Anwendungsfall angepasst werden, ohne dass durch eine vorherige Auswahl an Methoden der Aktionsbereich eingeschränkt wird.

4.3 Schnittstelle zwischen Workflow und Visualisierung

Wird der Workflow ausgeführt liefert dieser Daten. Diese sollen für die Visualisierungskomponente bereitgestellt werden.

Die Eclipse RCP nutzt das OSGi-Framework zur Modularisierung der RCP. Hierbei ist ein OSGi-Bundle (Plugin) die kleinste Einheit innerhalb der RCP. Ein solches Bundle kann ei-

genständig installiert und genutzt werden und wird über eine Service Registry im OSGi-Framework registriert. Andere Bundles können dann mithilfe der Metadaten in der Manifest-Datei über diese Service Registry Abhängigkeiten untereinander spezifizieren.

Diesen Mechanismus mit Services und Service Registry soll auch für die Interaktion zwischen dem Workflow-Editor und der RCE-Visualisierung genutzt werden. Die Daten eines Workflows sind komponentenspezifisch, das heißt jede Komponente liefert unterschiedliche Arten von Daten. Um diese Daten in der Visualisierung nutzen zu können, müssen die Daten in der Service Registry registriert werden und als Service bereitgestellt werden. Die Daten können dann von den anderen Bundles, wie dem Datenmanagement und auch von der Visualisierungskomponente abgegriffen werden.

Die Visualisierungskomponente stellt dann ein geeignetes Interface zur Verfügung, in dem die unterschiedlichen Datentypen definiert sind. Der Workflow liefert in erster Linie Punkte, Vektoren und Matrizen aus numerischen Datentypen (Integer, Double, Float, usw). Dazu könnten noch Funktionen in Betracht kommen (als String-Daten) und spezielle Dateiformate wie CPACS-Dateien, die nur von speziellen Zusatzprogrammen korrekt und vollständig visualisiert werden können.²¹

4.4 Separates Visualisierungsfenster vs. Visualisierung mit gegebenen Eclipse-RCP-Mechanismen

Dieses vorgestellte Visualisierungskonzept öffnet - nach dem Erstellen und Ausführen des Workflows - das separate Visualisierungsfenster um die Simulationsdaten (zur Laufzeit oder Nachbearbeitung) zu visualisieren. Aus der Anforderungsanalyse geht explizit hervor, dass RCE-Benutzer die Visualisierung in einem separaten Fenster bevorzugen. Betrachtet man jedoch die Eclipse RCP und insbesondere die darauf aufgebaute RCE-Software, so zeigt sich, dass eine Umsetzung mit den vorgegebenen Erweiterungsmechanismen der Eclipse RCP sinnvoll und vorteilhafter sein kann.

Um dies aufzulösen, muss zunächst geklärt werden, woher die Forderung nach einem separaten Fenster stammt und welchen Vorteil aus Sicht der RCE-Benutzer diese gegenüber einer Umsetzung mit den Eclipse Extensions und Extension Points bringen wird.

Viele der RCE-Benutzer haben vor der RCE-Entwicklung ModelCenter PHX als Simulations-

²¹vgl. Hall u.a. (2011): OSGi in Action. Stand: August 2012

framework genutzt und sind mit der Anwendung und der dazugehörigen Daten-Visualisierung vertraut. Da die RCE-Software keine Visualisierung besitzt, orientieren sich die ehemaligen ModelCenter-Nutzer natürlicherweise an der bestehenden ModelCenter-Visualisierung.

Die Visualisierung in der ModelCenter-Software geschieht komponentenweise. Hierzu wird bei der Komponentenvisualisierung zunächst die Komponente ausgeführt und danach die anfallenden Daten in den Visualisierungsbereich übertragen. Dieser Visualisierungsbereich befindet sich in einem eigenen Fenster. Da die RCE-Nutzer mit der Eclipse-RCP und deren Mechanismen und Möglichkeiten im Gegensatz zur ModelCenter-Software weniger vertraut sind, ist anzunehmen, dass die Anforderung nach einer Visualisierung in einem separaten Fenster daher kommt.

Zudem kann mit separaten Fenstern Workflows und Komponenten untereinander verglichen werden. Dieser explizite Anwendungsfall müsste daher in einer in RCE integrierten Umsetzung berücksichtigt werden.

Aus Entwicklersicht bietet eine Umsetzung mit den gegebenen Möglichkeiten viele Vorteile gegenüber einer Umsetzung in einem separaten Fenster. Aus diesem Grund werden im folgenden Kapitel diese beiden Möglichkeiten für eine RCE-Visualisierung prototypisch umgesetzt. Diese beiden Prototypen sollen dann - mit zusätzlichen Anmerkungen und Kommentaren - an die Nutzer gegeben werden. Dadurch wird den Nutzern zum einen eine weitere Möglichkeit für eine Integration der Visualisierung gezeigt, und zum anderen das Bewusstsein für die von der Eclipse RCP vorgesehenen Mechanismen erweitert.

5 Prototypische Implementierung der Visualisierungsoberfläche zu Vorführrzwecken

Dieses Kapitel beinhaltet die praktische Komponente der Bachelorarbeit. Für das im vergangenen Kapitel vorgestellte Konzept und die darin angesprochene Überlegung zur Art und Weise der Integration der Visualisierung in die RCE-Oberfläche ist hier nun die Umsetzung der Visualisierung zu finden. Anschließend werden die Vor- und Nachteile der Umsetzung zum einen als ausgegliedertes Fenster und zum anderen als in die RCE-integrierte Visualisierung mit Perspektive, Editoren und Views (s. hierzu Kapitel 2.2 'Übersicht über die RCE-Software') diskutiert.

Um den RCE-Benutzern eine weitere Möglichkeit der Realisierung einer Datenvisualisierung in RCE aufzuzeigen und das Erweiterungsprinzip der Eclipse RCP mit Extensions und Extension Points, Editors und Views näher zu bringen, sind beide Ansätze als testbare Oberflächen realisiert. Zum besseren Verständnis einiger Funktionen und Überlegungen sind diese Oberflächen mit Anmerkungen und Dokumentation versehen. In einem nächsten Schritt sollen diese beiden prototypisch realisierten Oberflächen an die RCE-Nutzer gegeben werden um zunächst von den Nutzern eine Rückmeldung zur integrierten Variante zu bekommen. Ist den Nutzern die Möglichkeit einer integrierten Visualisierung mit RCP-Mechanismen bekannt, kann über die Art und Weise der Integration der RCE-Visualisierern entschieden werden.

Die Anmerkungen und Kommentare zu den beiden Prototypen ist in diese integriert. Hierzu gibt es in der Eclipse RCP bereits eine integrierte Hilfe, die in jeder Applikation auf Basis der RCP eine Hilfedokumentation generiert. Diese generierte Dokumentation wird im RCP-Prototyp dazu genutzt, zu jeder View und Editor in der Visualisierungsperspektive (s. hierzu Kapitel 5.2 'RCE-Visualisierung mit Eclipse-RCP integrierter Oberfläche') Anmerkungen, Besonderheiten und Kommentare zur Aufmachung zu geben. Diese Hilfe befindet sich an der rechten Seite des RCP-Prototyps.

Aus Konsistenzgründen wird in der Umsetzung im von der RCE-Applikation losgelösten Prototyp am rechten Rand ebenfalls Kommentare und Anmerkungen zur Umsetzung und den Funktionen zu finden.

Die beiden Prototypen sind im folgenden detailliert dargestellt.

5.1 RCE-Visualisierung mit ausgegliederter Oberfläche

Der Prototyp mit ausgegliederter Oberfläche für die RCE-Visualisierung ist als Oberfläche mittels Swing-Komponenten realisiert. Dabei wird das in Kapitel 4 vorgestellte Visualisierungskonzept mit den einzelnen entworfenen Elementen umgesetzt.

Nach dem Visualisierungskonzept gibt es vier grundlegende Elemente, die in der Oberfläche wieder zu finden sein sollten: die Datentabelle, der Diagramm-/Kurvenbereich, die Werkzeug- und Funktionenpalette und eine Palette mit den unterschiedlichen Visualisierungsformen. Der Prototyp mit ausgelagerter Benutzeroberfläche und die genannten Grundelemente sind auf dem Screenshot in Abbildung 9 wiederzufinden.

Im Screenshot auf der Abbildung 9 sind links die Visualisierungsoberfläche und rechts einer der Funktionalitätendialoge dargestellt. Der Diagramm- und Kurvenbereich bildet den Kern der RCE-Visualisierung und beinhaltet die eigentliche Visualisierung. An die ModelCenter Software angelehnt, befindet sich dieser mittig im Visualisierungsfenster und kann neben der eigentlichen Visualisierung auch weitere Informationen zur Darstellung enthalten. Am linken Rand der Visualisierungsoberfläche befindet sich die Werkzeugleiste. Je nach Funktionsbereich öffnet man von dort aus die Dialoge, die dann die entsprechenden Funktionen beinhalten. Beispielhaft ist der Mathematik-Dialog im rechten Teil der Abbildung 9 zu finden. Die Auswahl an Visualisierungsarten ist am rechten Rand der Oberfläche zu finden. Je nach gewählter Visualisierungsart öffnet sich dann die jeweilige Visualisierung. Im Einstellungen-Dialog (Settings) kann dann die jeweilige Anzeige angepasst werden. Zuletzt ist am unteren Rand die Datentabelle zu finden.

Die ausgegliederte Visualisierungsoberfläche öffnet sich, sobald ein Workflow ausgeführt wird. Dabei wird je Fenster nur die Daten einer einzelnen Komponente angezeigt (bzw. der gesamte Workflow wird nach außen hin gekapselt, ähnlich ModelCenter). Mehrere geöffnete Visualisierungsfenster neben der eigentlichen RCE-Software zur gleichen Zeit sind somit möglich und gewährleisten die Möglichkeit des Vergleichens von Workflows, Komponenten und Iterationsschritten.

Es handelt sich hierbei zunächst nur um einen der beiden Prototypen für die RCE-Oberfläche. Dieser dient dazu, herauszufinden, ob es seitens der RCE-Benutzer weitere Vorstellungen und Einschränkungen für die Oberfläche der RCE-Visualisierung gibt. Hierzu befinden sich in diesem Prototyp Anmerkungen und Kommentare in Form einer Hilfeleiste am rechten Rand. Die Anmerkungen dienen zum einen dazu, wichtige Design-Entscheidungen kurz zu erläutern und einzelne Funktionen und umgesetzte Elemente erklärend zu ergänzen. Die Kommentare und Erklärungen für Funktionalitäten ähneln weitestgehend denen des anderen Prototyps, da beide auf dem gleichen Konzept aufbauen. Im integrierten Prototyp wird lediglich die bereits bestehende dafür vorgesehene Hilfefunktion der Eclipse RCP genutzt (s. Kapitel 5.2).

5.2 RCE-Visualisierung mit Eclipse-RCP integrierter Oberfläche

Dieser zweite Prototyp ist eine alternative Möglichkeit für die Umsetzung der RCE-Visualisierung. Hierzu sind die einzelnen Elemente des Visualisierungskonzeptes den bestehenden

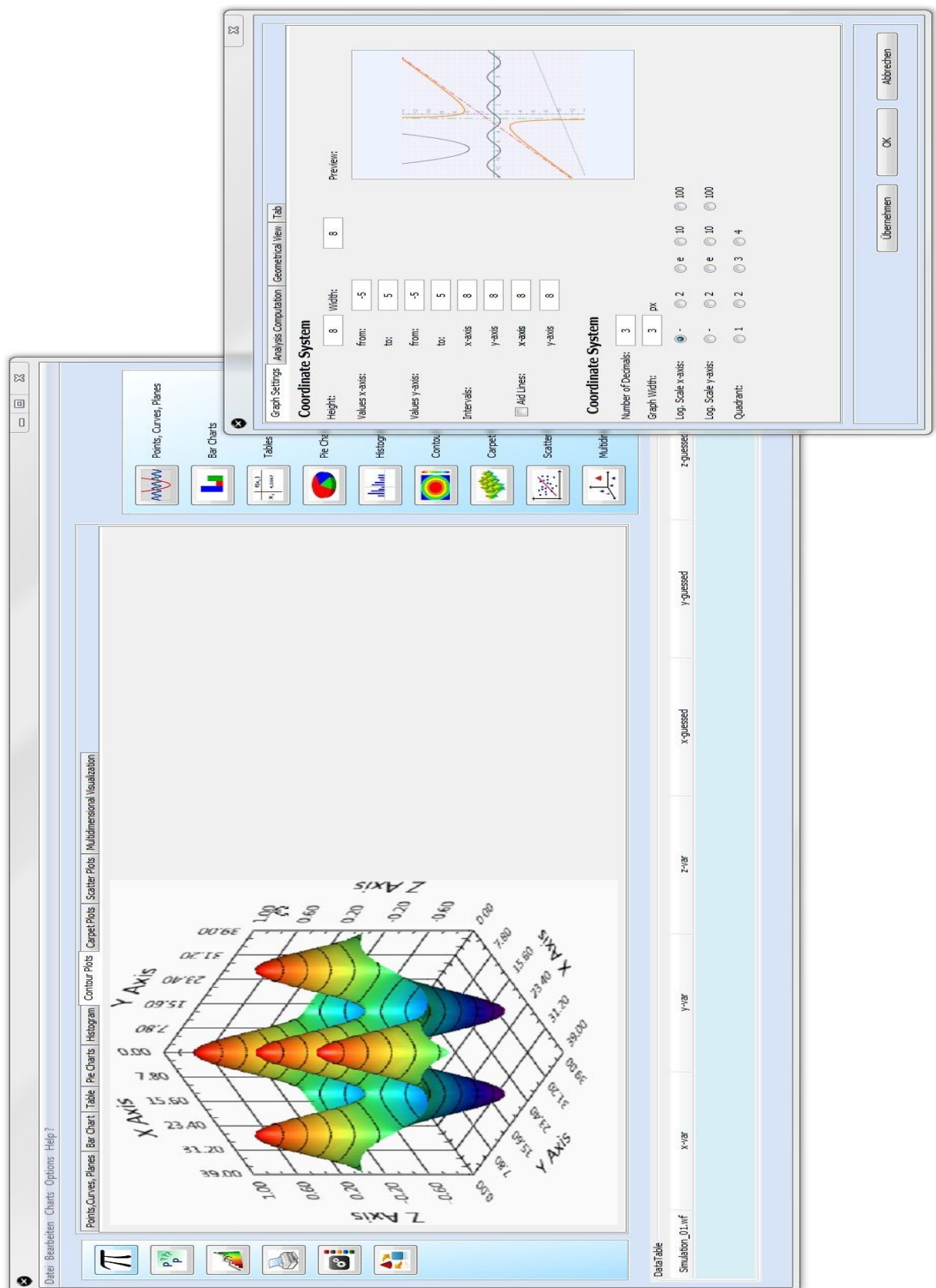


Abbildung 9: Screenshot der ausgegliederten Oberfläche

Eclipse RCP-Mechanismen zugeordnet.

Die Eclipse RCP bietet als Basisplattform bereits einen umfangreichen Mechanismus zur Erweiterung der Oberfläche und der Funktionalitäten. Zunächst kann die Eclipse Plattform über Plugins beliebig erweitert werden. Einem Plugin kann dabei über sogenannte Extension Points Funktionalität zugewiesen werden. In der Eclipse RCP gibt es zusätzlich definierte Oberflächenelemente, wie Editors, Views und Perspektiven (s. Kapitel 2.2). Diese nutzen ebenfalls den Extension Point-Mechanismus und sind somit gut geeignet für jegliche Arten von RCE-Erweiterungen.²²

Die RCP besteht gänzlich aus Plugins, weswegen Funktionalitäten an den unterschiedlichsten Stellen angebracht werden können. Zudem können auch eigene Plugins als Extension Points und Extensions fungieren.

Dieser Prototyp zielt darauf ab, die gegebenen Mechanismen und Elemente der RCP möglichst effizient und sinnvoll auszuschöpfen. Aus diesem Grund sind die Elemente des Visualisierungskonzeptes den bereits definierten Oberflächenelementen der RCP zugeordnet.

Wird ein Workflow ausgeführt, ändert sich der Anwendungsbereich der RCE-Oberfläche: Die Erarbeitung der Simulation ist abgeschlossen und wird nun überwacht und im Anschluss ausgewertet (s. hierzu Abbildung 7). Dafür wird dann auch von der RCE-Perspektive in die Visualisierungsperspektive der RCE-Anwendung gewechselt. Diese beinhaltet alle weiteren relevanten Elemente für die Visualisierung in RCE. Der Prototyp der Visualisierungsperspektive ist in der Abbildung 10 zu sehen.

Der Kern der Visualisierung, der Diagramm- und Kurvenbereich, ist in diesem Ansatz als Visualisierungseditor angedacht. Editoren der RCP werden genutzt, um viele Daten einer bestimmten Art oder eines Datentyps darzustellen und sind so gut geeignet zur Visualisierung von Workflow-Daten. Dieser Editor ist im Prototyp zunächst nur als Platzhalter zu sehen (Abbildung 10, Mitte), da für die Definition eines (RCE-eigenen) Visualisierungseditors bestimmte Fragen zur Schnittstelle zum Workflow-Editor (s. Kapitel 4.3) geklärt sein müssen.

An der rechten Seite befindet sich die Visualisierungspalette. Diese dient vornehmlich der Auswahl der gewünschten Visualisierungsart, und beinhaltet zudem die Historie der zuletzt benutzten Aktionen. An der linken Seite befindet sich die Workflow View. Sie repräsentiert

²²Extension (engl.) = Erweiterung

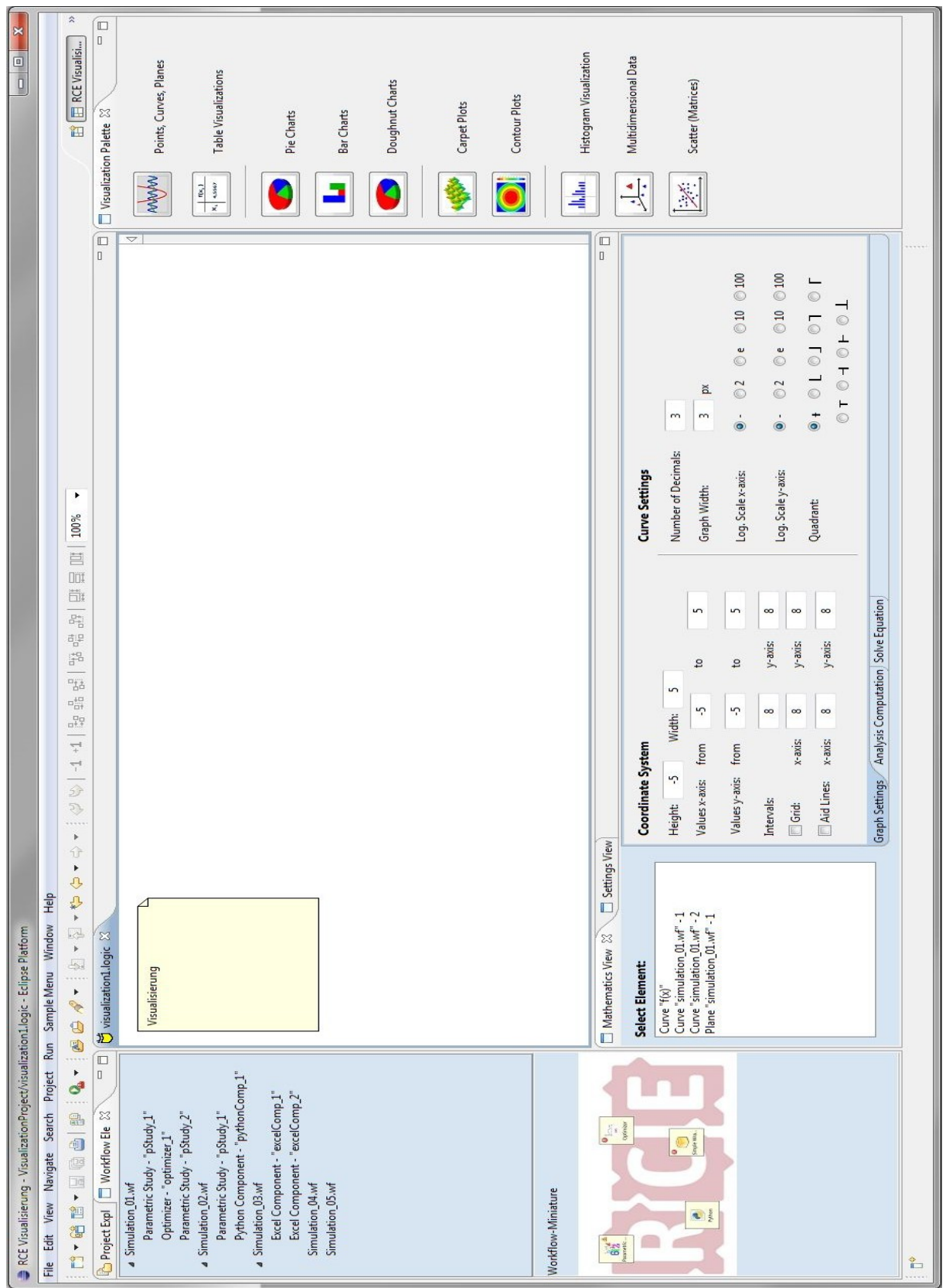


Abbildung 10: Screenshot der Umsetzung der RCE-Visualisierung mit integrierter Oberfläche

den Workflow-Editor und bietet die Möglichkeit, andere Komponenten zur Visualisierung auszuwählen. Der Workflow wird hier als hierarchischer Baum und als Miniaturansicht bereitgestellt.

Am unteren Rand sind die Views für die jeweiligen Funktions- und Anwendungsbereiche zu finden. Typische Funktionsbereiche sind zum Beispiel mathematische Funktionen, Funktionen zur statistischen Auswertung, für den Import und Export von Daten oder zur Veränderung der Visualisierung. Diese werden jeweils in einer eigenen View gesammelt und in der Oberfläche zur Verfügung gestellt. Zusätzlich können spezielle Anwendungsfälle als Views abgedeckt werden. Ein solcher spezieller Anwendungsfall ist beispielsweise der Vergleich von mehreren Komponenten, Iterationen und Workflows miteinander. Dieser Fall entstammt der Befragung der RCE-Benutzer und ist als solcher bereits berücksichtigt. Ein weiterer Anwendungsfall ist der sogenannte Prediction Profiler, mit dem Daten einer Simulation in Abhängigkeit unterschiedlicher Variablen abgeschätzt werden können.

Die RCP bietet für Editoren und dazugehörige Views bereits einen Kommunikationsmechanismus, mit dem es möglich ist, eine Auswahl im Visualisierungseditor abzufangen und die dazugehörigen Daten in den jeweiligen Views bereitzustellen. Dieser Mechanismus muss so nicht erneut implementiert werden.

Zu Demonstrationszwecken wird in diesem vorgestellten Prototyp ein weiterer RCP-Mechanismus verwendet, die Hilfefunktionalität. Diese zeigt zum jeweiligen ausgewählten Oberflächenelement, also zu jeder View, jedem Editor, etc. die entsprechende Seite der Hilfe an. So werden Anmerkungen und Erklärungen in der Oberfläche gezeigt und die Hilfe kann exportiert und zur Dokumentation genutzt werden.

5.3 Gegenüberstellung von ausgegliederter und integrierter RCE-Visualisierung

Im Folgenden werden nun die beiden vorgestellten Prototypen gegenübergestellt und deren Vor- und Nachteile erläutert. Das Ergebnis ist dann zusammenfassend in der Tabelle 5 zu sehen.

Für eine Umsetzung der Visualisierung im separaten Fenster spricht vor allem die explizite Anforderung durch die Benutzer. Jedoch ist anzunehmen, dass eine weitere Umsetzungsmöglichkeit bei den RCE-Nutzern nicht präsent ist und deshalb nicht in Betracht gezo-

gen worden ist.

Sowohl eine Umsetzung als separates Fenster, als auch eine in die Eclipse RCP integrierte Visualisierung, sollte über die gleiche Schnittstelle zur RCE-Software angebunden werden. Aus diesem Grund ist der Aufwand für die Schnittstelle in beiden Fällen gleich hoch. Jedoch ist abzusehen, dass die Nutzung von bereits definierten Erweiterungspunkten die Gestaltung und Umsetzung der Oberfläche wesentlich vereinfachen. Beispielsweise gibt es bereits Mechanismen für eine Kommunikation zwischen Editoren und Views, wohingegen die Kommunikation zwischen der RCP und dem integrierten Fenster zunächst noch nicht besteht. Auch die bereits integrierte RCP-Hilfe kann für RCP-Elemente einfach genutzt werden und die einzelnen Views können von dem Hauptfenster abgedockt werden.

Eine Umsetzung als separates Fenster mit SWT-Elementen scheint auf den ersten Blick mehr Möglichkeiten und kaum Einschränkungen zu bieten gegenüber einer Übertragung auf RCP-Elemente. Mit mehr Möglichkeiten besteht jedoch auch die Gefahr, dass eine intuitive, benutzerfreundliche Bedienung nicht oder nur schlecht umgesetzt wird und die RCE-Visualisierung sehr komplex wird.

Ein weiterer Vorteil der integrierten Umsetzung ist, dass die RCE-Visualisierung einfach zu erweitern ist und dabei nur die hinzugefügten Elemente auf Benutzbarkeit überprüft werden müssen. Die RCE-Perspektive kann beliebig zusammengestellt und abgespeichert werden. Die Komplexität der RCE-Visualisierung verändert sich deshalb beim Hinzufügen einer Komponente nicht.

Die Usability und intuitive Benutzbarkeit des Prototyps im separaten Fenster ist schwieriger zu gewährleisten, als bei einer Umsetzung als RCE-Perspektive mit Visualisierungseditor und passenden Views. Bei der Umsetzung als integrierte Visualisierung übernimmt bereits die RCP die grundlegenden Entscheidungen zu Layout, Design und Aufmachung. Der Entwickler muss sich kaum Gedanken über die generelle Benutzbarkeit mehr machen. Aus der Wartbarkeit bei Updates der RCP oder Hinzufügen von weiteren Visualisierungselementen gestaltet sich durch die definierten Eclipse RCP Mechanismen einfacher.

Das schwerwiegendste Problem bei einer Visualisierung in einem separaten Fenster besteht jedoch in der Inkonsistenz zwischen der Eclipse RCP mit der darauf aufbauenden RCE-Software und dem Visualisierungsfenster. Die RCE-Software baut als Rich Client auf der Eclipse RCP auf und ist durchgehend konsistent zur Basisplattform. Bereits die RCE-Software benutzt

ausschließlich die bereitgestellten Mechanismen und bildet zusammen mit der RCP eine gut durchdachte Einheit. Die Beibehaltung dieser Einheit soll in der weiteren Entwicklung der RCE-Software beibehalten werden. Die Erweiterung von RCE um eine Visualisierungsperspektive ist somit sinnvoll.

Zusammenfassend sind die Vor- und Nachteile der beiden Umsetzungen der Visualisierungsoberfläche in der folgenden Tabelle aufgeführt.

	Prototyp 'Separat'	Prototyp 'Integriert'
Entspricht Anforderungen	✓	X
Aufwand		
Oberfläche	X	✓
Schnittstelle	○	○
Funktionalität		
Vordefinierte Mechanismen	X	✓
Möglichkeiten/Grenzen	✓	X
Erweiterbarkeit	X	✓
Benutzerfreundlichkeit		
Ähnlichkeit zu ModelCenter	✓	X
Intuitiv benutzbar	○	✓
Konsistenz u. Einfachheit	X	✓
Komplexität	X	✓
Wartbarkeit		
Beibehaltung der Benutzbarkeit	X	✓
Updates	X	✓
Konsistenz		
zur Eclipse RCP	X	✓
zu RCE	X	✓
<div> <div>✓ Vorteil</div> <div>X Nachteil</div> <div>○ Neutral</div> </div>		

Tabelle 5: Übersicht über Vor- und Nachteile der vorgestellten Prototypen

Nimmt man die diskutierten Punkte als Grundlage für eine Bewertung der Umsetzungsmöglichkeiten, überwiegen die Vorteile, die eine Visualisierung von Workflow-Daten in einer Visua-

lisierungsperspektive innerhalb der RCE-Software mit sich bringt. Auch wenn beide Umsetzungen an der gleichen Schnittstelle angehängt werden, bleibt ein nicht integriertes separates Fenster ein optischer Fremdkörper innerhalb der RCE-Software. Die Konsistenz innerhalb eines Softwareproduktes und die Benutzerfreundlichkeit sind Kriterien, die die Qualität eines Produktes definieren. Aus diesem Grund ist eine Umsetzung der Visualisierung mit den gegebenen RCP-Mechanismen vorzuziehen.

6 Fazit

Ziel dieser Bachelorarbeit ist es gewesen, durch das Hinzufügen der Datenvisualisierung die RCE-Software zu verbessern. Neben der Durchführung ist die Auswertung von Simulationen in der Wissenschaft von großer Bedeutung. Aus diesem Grund ist mithilfe dieser Arbeit der Arbeitsablauf der RCE-Software um den Schritt der Datenauswertung erweitert worden.

Für eine Entwicklung der RCE-Visualisierung sind zunächst Anforderungen an eine Datenvisualisierung zusammengetragen und im Anschluss ausgewertet, strukturiert und analysiert worden. Die Schwierigkeit bei der Informationsbeschaffung besteht darin, dass viele Anwender in ganz Deutschland verteilt sind und sich so ein flächendeckendes Gesamtbild nur schwer erschließen lässt. Anhand der strukturierten und analysierten Anforderungen ist dann ein Konzept mit notwendigen Elementen der Visualisierung entwickelt und Möglichkeiten der Realisierung betrachtet worden.

Während der Konzeptionierung ist die Frage nach der Art und Weise der Integration der Visualisierungsoberfläche in den Vordergrund gerückt. Aus Benutzersicht ist dabei die Visualisierung in einem separaten Fenster außerhalb der RCE-Software erforderlich. Dazu im Konflikt stehen die Einschätzungen der RCE-Entwickler, die eine konsistente Umsetzung mit den gegebenen Methoden der Eclipse RCP wünschen. Um diese Frage zu klären sind anhand des erarbeiteten Konzeptes beide in Betracht kommenden Oberflächen prototypisch umgesetzt worden.

Anhand dieser Prototypen wird im nächsten Schritt eine Entscheidung zur Aufmachung der Visualisierung getroffen. RCE lebt von der engen Interaktion zwischen den Entwicklern und Benutzern, weshalb es wichtig ist, die Benutzer in die Entscheidung nach der Oberfläche miteinzubeziehen. Ist eine Entscheidung getroffen, kann im Anschluss strukturiert und planmäßig mit der Implementierung der Visualisierung begonnen werden. Die Visualisierungskomponente bietet enormes Potential für eine Verbesserung der Softwarequalität von RCE aus Benutzersicht, da Veränderungen vor allem an der Benutzeroberfläche zu erkennen sein werden und sowohl für die Nutzer, als auch für die Entwickler ein neuer Anwendungsbereich von RCE hinzukommt.

Mit der Bachelorarbeit ist so eine gut geplante Basis für die Implementierung geschaffen worden, anhand der eine strukturierte und effiziente Umsetzung der RCE-Visualisierung vor-

genommen werden kann.

Die RCE-Visualisierung wird einen gewichtigen Anteil an der gesamten RCE-Oberfläche haben und soll somit zur Verbesserung der RCE-Software beitragen. Zudem ist die Visualisierung mehrfach von Anwendern nachgefragt und bisher nur unzureichend bereitgestellt worden. Mit dieser Arbeit ist ein neues Kapitel der RCE-Software mit einer guten Planungs- und Konzeptionsphase eingeleitet worden. Durch eine vorbildliche, strukturierte Vorgehensweise und gut ausgewerteter und zusammengefasster Dokumentation steht der RCE-Visualisierung nichts mehr im Weg. Somit hat diese Arbeit maßgeblich zur Erweiterung und Verbesserung der RCE-Software und deren Anwendungsbereichen beigetragen.

Literatur

- DLR (2011): Offizielle Webpräsenz des Deutschen Zentrums für Luft- und Raumfahrt e.V., Website, URL: <http://www.dlr.de/dlr>, abgerufen am 04. Juni 2012.
- ECLIPSE FOUNDATION (2011): Eclipse Documentation, URL: <http://help.eclipse.org/indigo/index.jsp>, abgerufen am 23. Juli 2012.
- ECLIPSE FOUNDATION (2012a): Eclipsepedia, URL: <http://wiki.eclipse.org>, abgerufen am 25. Juni 2012.
- ECLIPSE FOUNDATION (2012b): GEF(MVC), URL: http://www.eclipse.org/gef/gef_mvc/index.php, abgerufen am 9. August 2012.
- FRAUNHOFER SCAI (2012): Offizielle Webpräsenz des Fraunhofer-Instituts für Algorithmen und Wissenschaftliches Rechnen, Website, URL: <http://www.scai.fraunhofer.de/produkte.html>, abgerufen am 25. Juni 2012.
- FRY, B. (2008): Visualizing Data, erste auflage Aufl., O'Reilly Media, Inc., Sebastopol, California.
- HALL, R. S./PAULS, K./McCULLOCH, S./SAVAGE, D. (2011): OSGi in Action, Manning Publications Co., Stanford, Connecticut.
- HEGE, H.-C./POLTHIER, K. (2010): Mathematical Visualization. Algorithms, Applications and Numerics, erste auflage Aufl., Springer-Verlag, Berlin, Heidelberg.
- HILLE, H. (2009): Eclipse Rich Client Platform (RCP): Einführung in eine komponentenbasierte Programmierung mit Eclipse, erste auflage Aufl., Europäischer Hochschulverlag GmbH Co. KG, Bremen.
- INTERNET ENGINEERING TASK FORCE (IETF) (2005): RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files, URL: <http://tools.ietf.org/html/rfc4180>, abgerufen am 9. August 2012.
- PHOENIX INTEGRATION (1999-2011): PHX ModelCenter Hilfe, abgerufen am 23. Juli 2012.
- PHOENIX INTEGRATION (2012): PHX ModelCenter - Desktop Trade Studies, Website, URL: <http://www.phoenix-int.com/software/phx-modelcenter.php>, abgerufen am 17. Juli 2012.

- QUATRANI, T. (1998): Visual Modeling with Rational Rose and UML, vierte auflage Aufl., Addison Wesley Longman, Inc., Reading, Massachusetts.
- SHNEIDERMAN, B. (1996): The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, Conference Publication, University of Maryland, Department of Computer Science, Human-Computer Interaction Laboratory, and Institute for Systems Research, This paper appears in: IEEE Symposium on Visual Languages.
- SILVERLIGHT, M. (2012): PivotViewer, Website, URL: <https://www.microsoft.com/silverlight/pivotviewer/>, abgerufen am 26. Juli 2012.
- WINOGRAD, T./STANFORD UNIVERSITY AND INTERVAL RESEARCH CORPORATION (1996): Bringing Design to Software, ACM Press, New York.
- WIRDEMAN, R. (2011): SCRUM mit User Stories, zweite erweiterte auflage Aufl., Carl Hanser Verlag, München, Wien.
- WOLFRAM RESEARCH (2012): Offizielle Webpräsenz von Wolfram Research, Website, URL: <http://www.wolfram.com/mathematica/>, abgerufen am 26. Juli 2012.
- WÜTHERICH, G./HARTMANN, N./KOLB, B./LÜBKEN, M. (2008): Die OSGi Service Plattform, erste auflage Aufl., dpunkt.verlag GmbH, Heidelberg.

Anhang

A. Fragebogen zur Anforderungsanalyse

Benutzer-Anforderungen an Visualisierung von Daten in RCE

Allgemeine Anforderungen

- Welche Visualisierungen sind im RCE-Kontext allgemein anzustreben? (Graphen, Diagramme, Tabellen, 3D-Graphen, ...)
- Wie ist die Visualisierung der RCE-Daten am sinnvollsten? (als externes Programm, in RCE integriert, in verschiedenen Fenstern, als „Visualisierungskomponente, ...)
- Wann ist eine Darstellung der Daten gewünscht? (zur Laufzeit, nach der Workflow-Ausführung, erwarteter Output,...)
- Welche Daten sollen dargestellt werden? (Komponenten, Input/ -Output, Workflow, ...)
- Welche Komponenten/Anwendungsfälle könnten es in Zukunft geben und welche Darstellungsformen sind dabei von Interesse?
- Welchen Vorteil bietet RCE gegenüber anderen Software-Produkten, beispielsweise ModelCenter?

Komponenten betreffende Anforderungen

- Welche Daten stehen bei welcher Komponente im Mittelpunkt und sind primär von Interesse? (Parametric Study –Graphen, Optimierer - Korrelationsdiagramme, ...)
- Was ist überflüssig/wünschenswert?
- Wie sehen die zu visualisierenden Daten aus und welche Darstellungsform sollte die Standard-Darstellung sein?

Anforderungen an Funktionen

- Welche Funktionalitäten sollten in der RCE-Software für die Visualisierung zur Verfügung stehen?
 - Vergleich von zwei Graphen (Graphen in einem/mehreren Koordinatensystemen, unterschiedliche Graphen je Iteration, ...)
 - Export-Funktionen
 - ...

B. Zusammenfassung über den Input der RCE-Benutzer zur Visualisierung

Neben den Standardvisualisierungsarten sind folgende Visualisierungsarten aus den Antworten der Benutzer herauszulesen: 2D und 3D Graphen, Carpet Plots, Contour Plots und der Prediction Profiler. Letzterer wird in der RCE-Umgebung als Anwendungsfall behandelt. Zudem ist sowohl die Visualisierung zur Laufzeit, als auch zur Nachbearbeitung gewünscht. Die zu visualisierenden Daten sind in der Regel Verläufe von Funktionswerten über Variablen, Punkte, also numerische Daten. Input und Output von Komponenten und Workflows sollen visualisiert werden, zudem Daten zu Anzahl und Zeit einer Komponente.

Als konkrete Anwendungsfälle stehen vor allem der Prediction Profiler und die Darstellung der Datenflüsse im Vordergrund. Zudem soll es eine Schnittstelle geben, mit der externe Daten visualisieren zu können.

Der Vergleich von Daten und Graphen unterschiedlicher Art in einem und mehreren Koordinatensystemen soll zudem möglich sein. Die Visualisierungen sollen als Grafiken und die Daten als CSV-Dateien exportierbar sein.

C. Übersicht über elektronische Anhänge

- Bachelorarbeit in elektronischer Form
- Ben Shneiderman.pdf: Conference Paper
- Quellcode der Prototypen